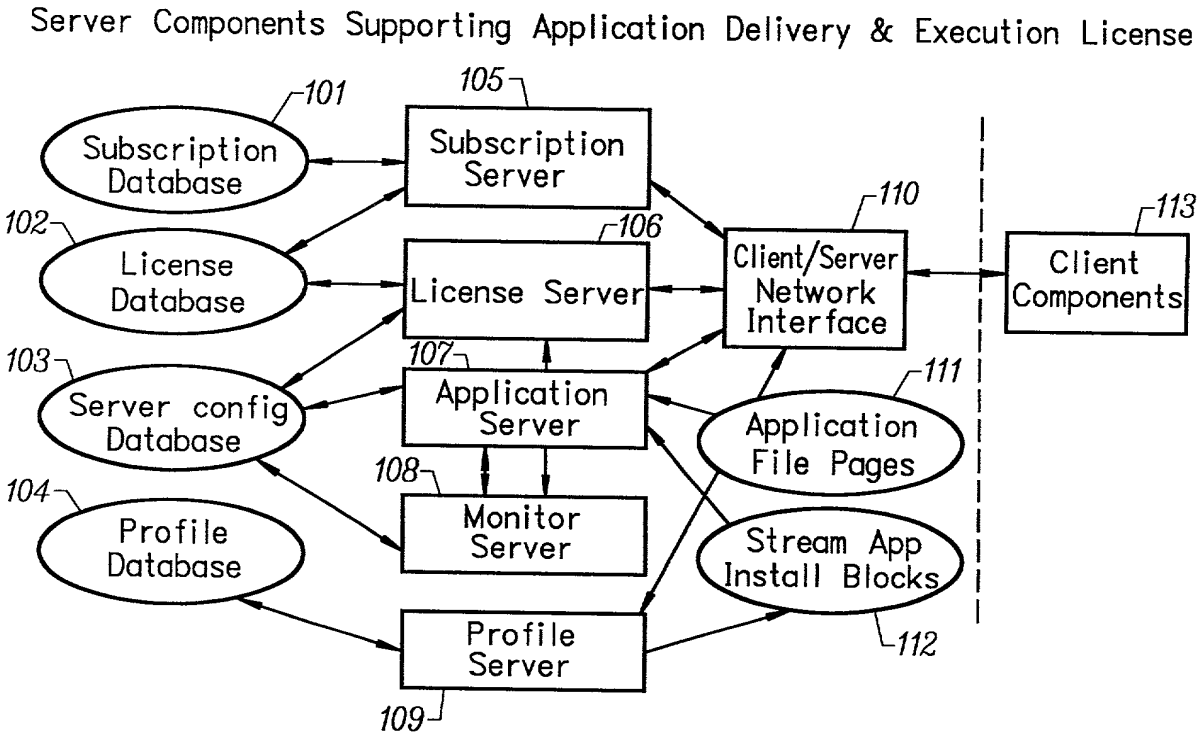

Server Components Supporting Application Delivery & Execution License
101
Subscription Database
105
Subscription Server
102
License Database
106
License Server
110
Client/Server Network Interface
113
Client Components
103
Server config Database
107
Application Server
111
Application File Pages
104
Profile Database
108
Monitor Server
Stream App Install Blocks
112
Profile Server
109

FIG. 1

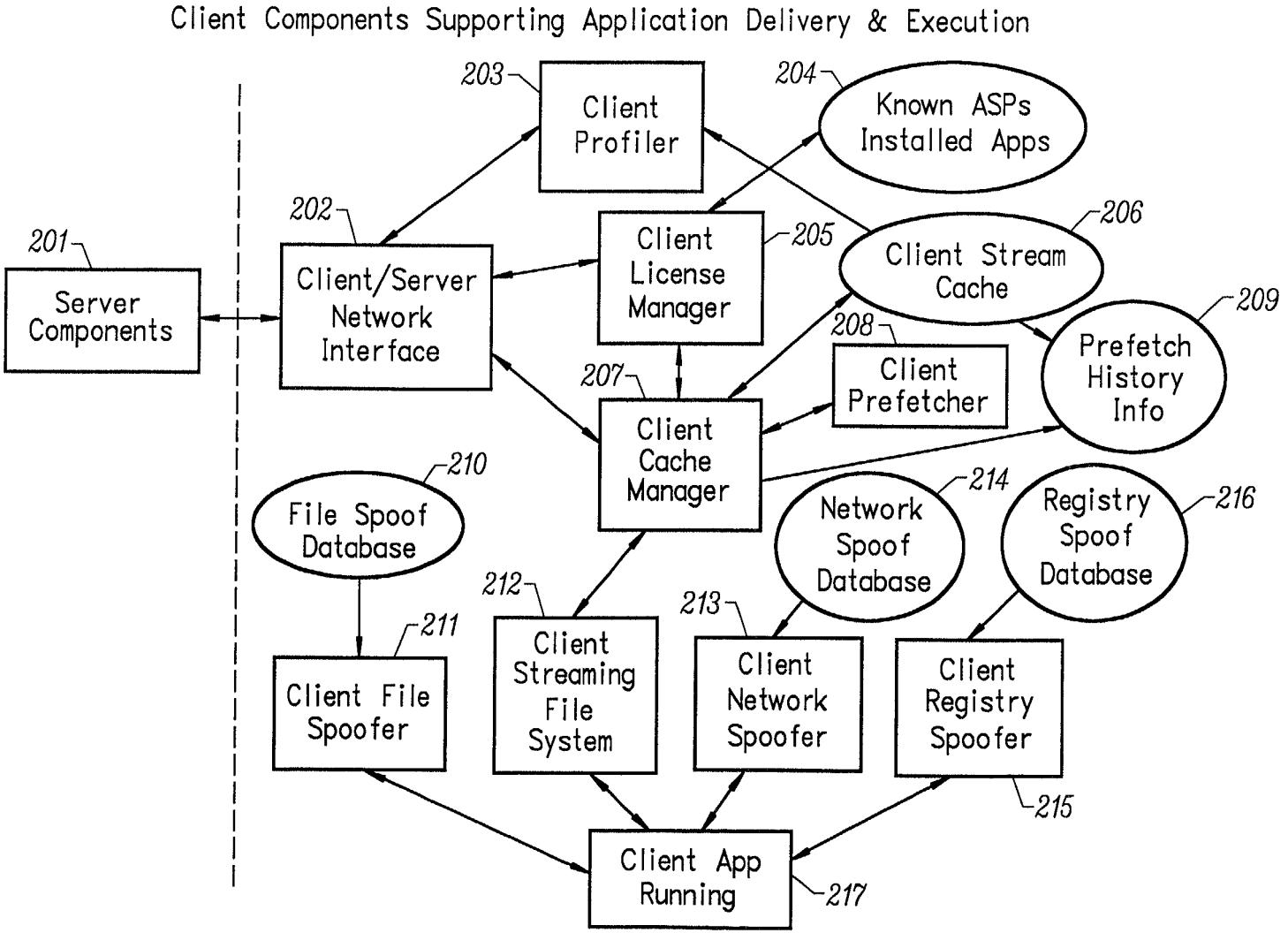

Client Components Supporting Application Delivery & Execution
203
Client Profiler
204
Known ASPs Installed Apps
202
201
Server Components
Client/Server Network Interface
Client License Manager
205
206
Client Stream Cache
209
208
Client Prefetcher
Prefetch History Info
207
Client Cache Manager
210
File Spoof Database
214
Network Spoof Database
216
Registry Spoof Database
212
213
211
Client File Spoofer
Client Streaming File System
Client Network Spoofer
Client Registry Spoofer
215
Client App Running
217

FIG. 2

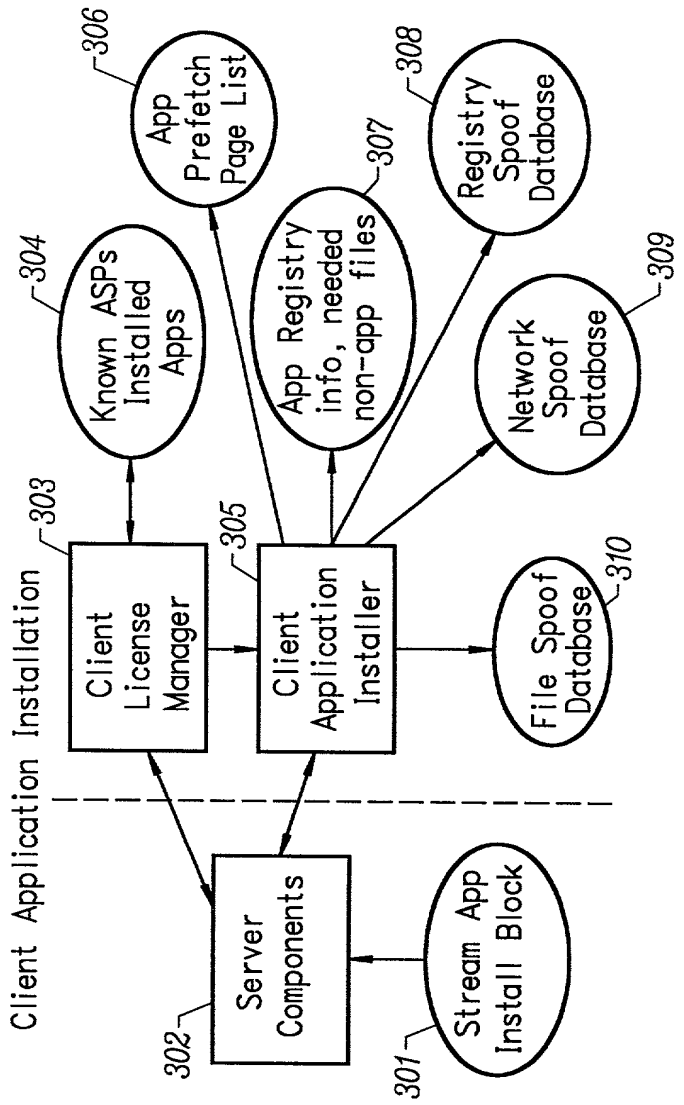

Client Application Installation
Client License Manager
303
Client Application Installer
305
Server Components
302
Stream App Install Block
301
Known ASPs Installed Apps
304
App Prefetch Page List
306
App Registry info, needed non-app files
307
Registry Spoof Database
308
Network Spoof Database
309
File Spoof Database
310

FIG. 3

Microsoft Confidential

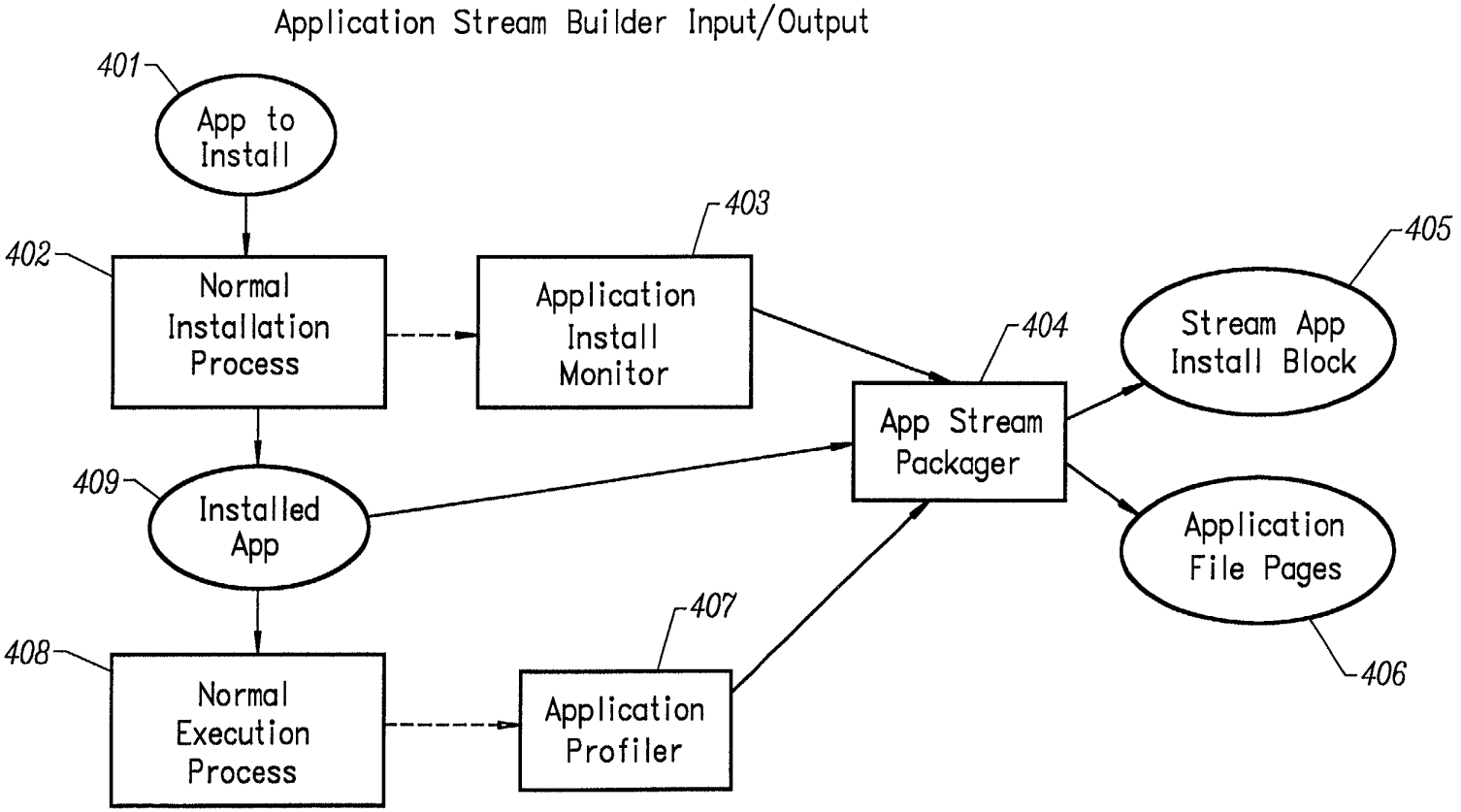
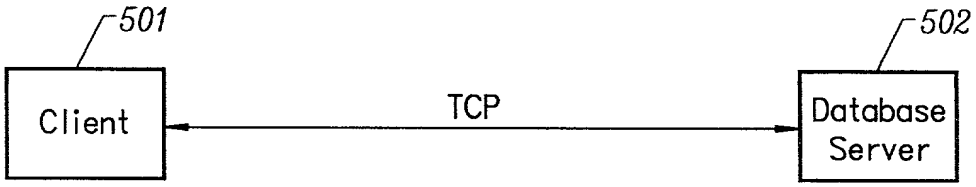
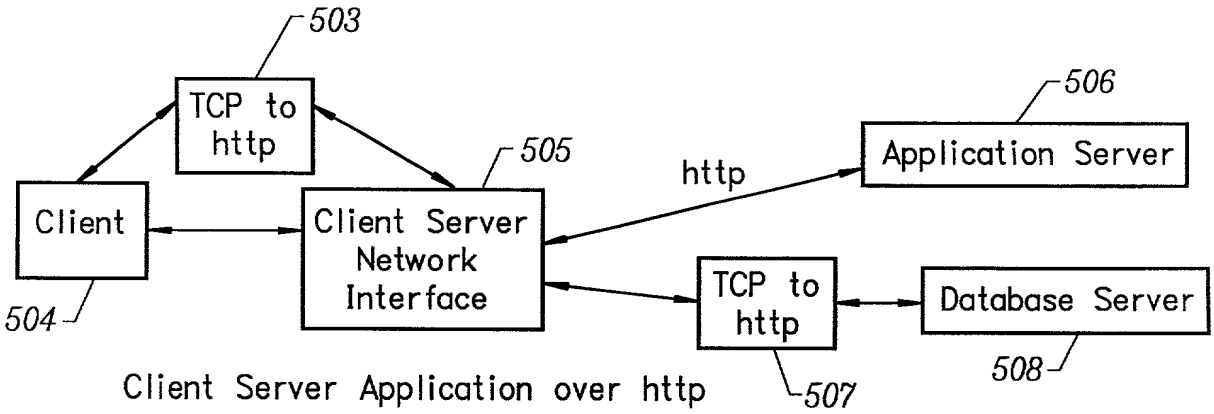


FIG. 4



Client-Server Application over TCP

FIG. 5A



Client Server Application over http

FIG. 5B

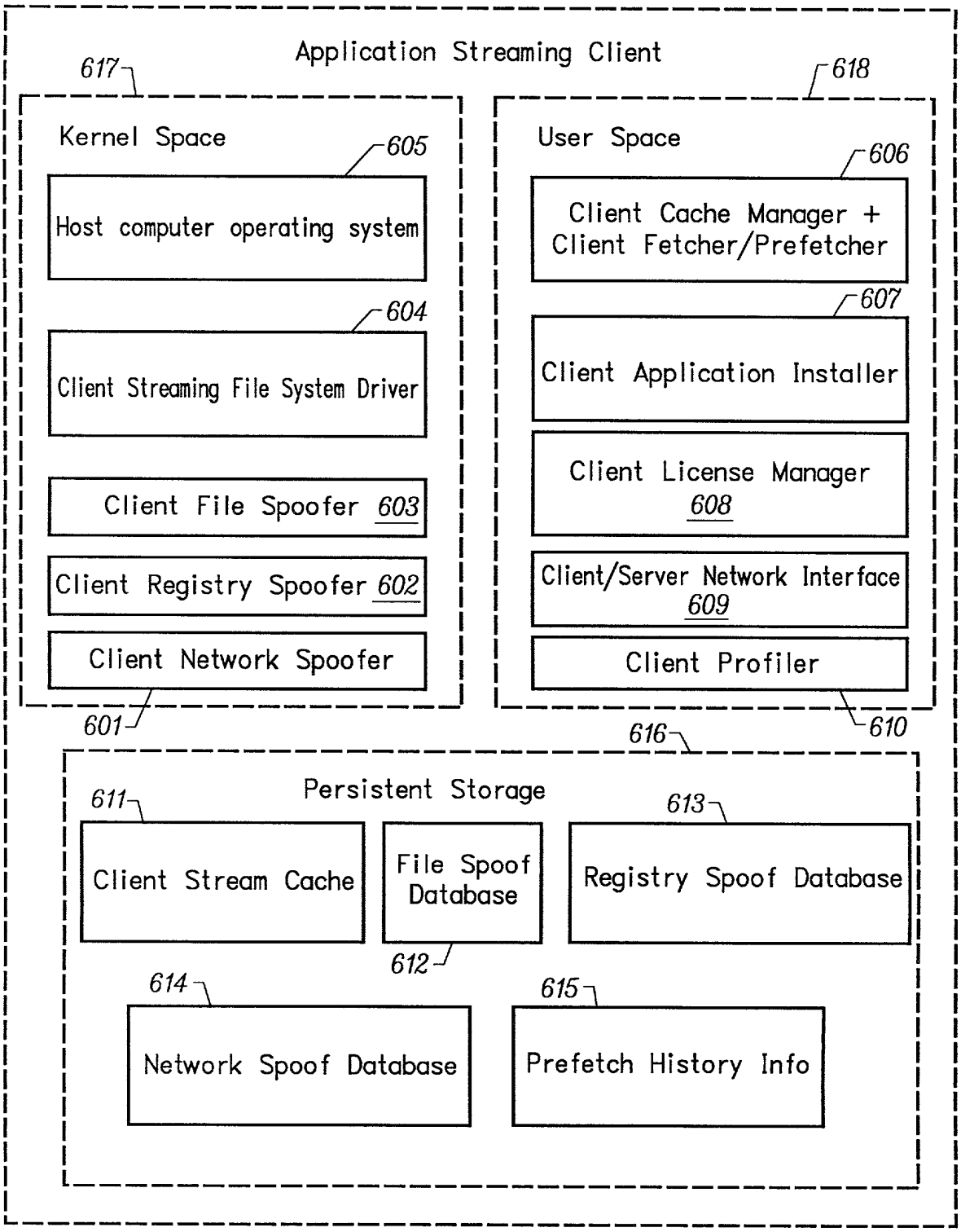

Application Streaming Client
617
618
Kernel Space
605
Host computer operating system
604
Client Streaming File System Driver
Client File Spoofer 603
Client Registry Spoofer 602
Client Network Spoofer
601
User Space
606
Client Cache Manager + Client Fetcher/Prefetcher
607
Client Application Installer
Client License Manager 608
Client/Server Network Interface 609
Client Profiler
610
616
Persistent Storage
611
Client Stream Cache
File Spoof Database
612
613
Registry Spoof Database
614
Network Spoof Database
615
Prefetch History Info

FIG. 6A

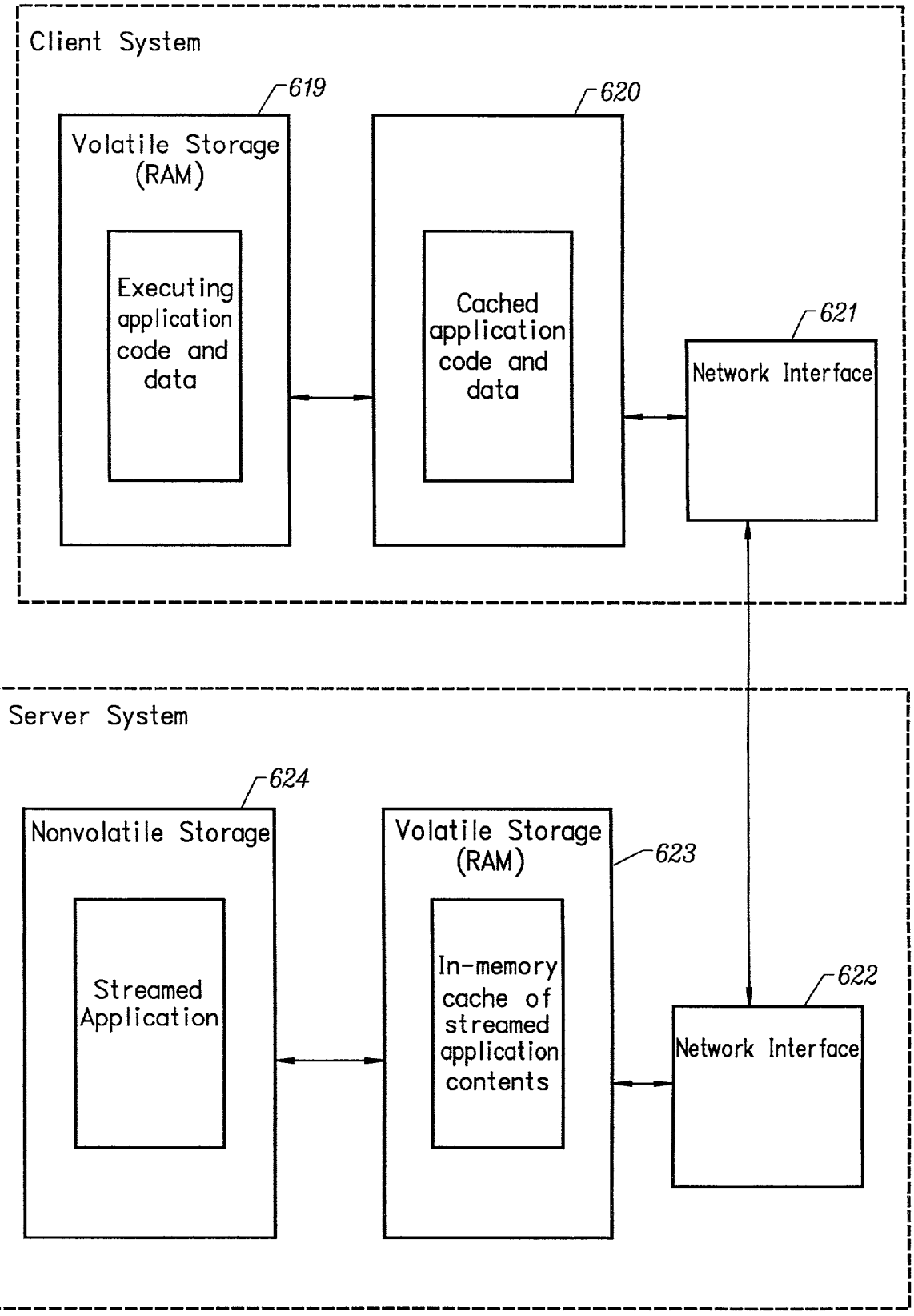

Client System
619
Volatile Storage (RAM)
Executing application code and data
620
Cached application code and data
621
Network Interface
Server System
624
Nonvolatile Storage
Streamed Application
Volatile Storage (RAM)
623
In-memory cache of streamed application contents
622
Network Interface

FIG. 6B

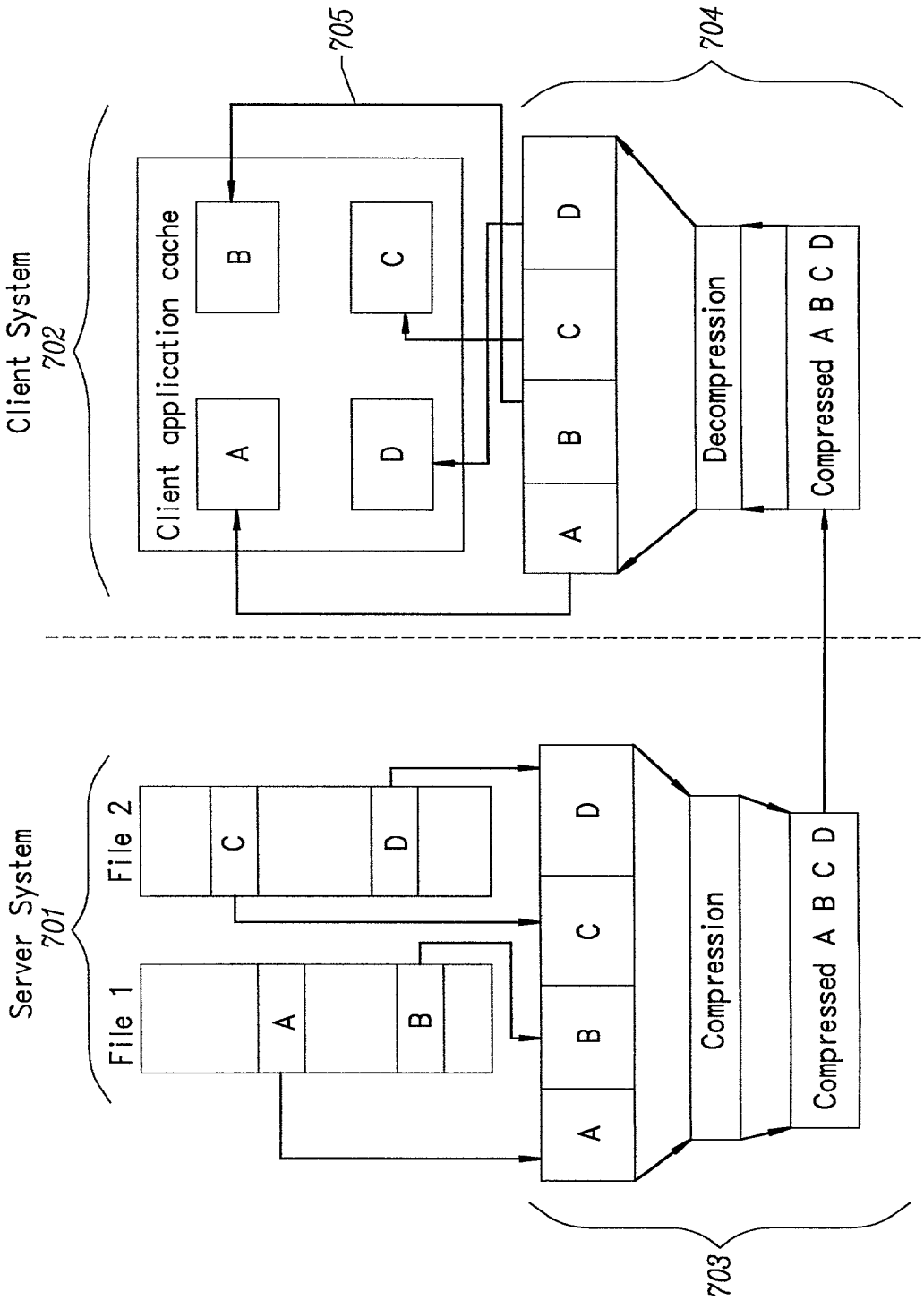

Server System
701
File 1
A
B
File 2
C
D
703
A
B
C
D
Compression
Compressed A B C D
705
Client System
702
Client application cache
A
B
C
D
704
A
B
C
D
Decompression
Compressed A B C D

FIG. 7A

Document Received

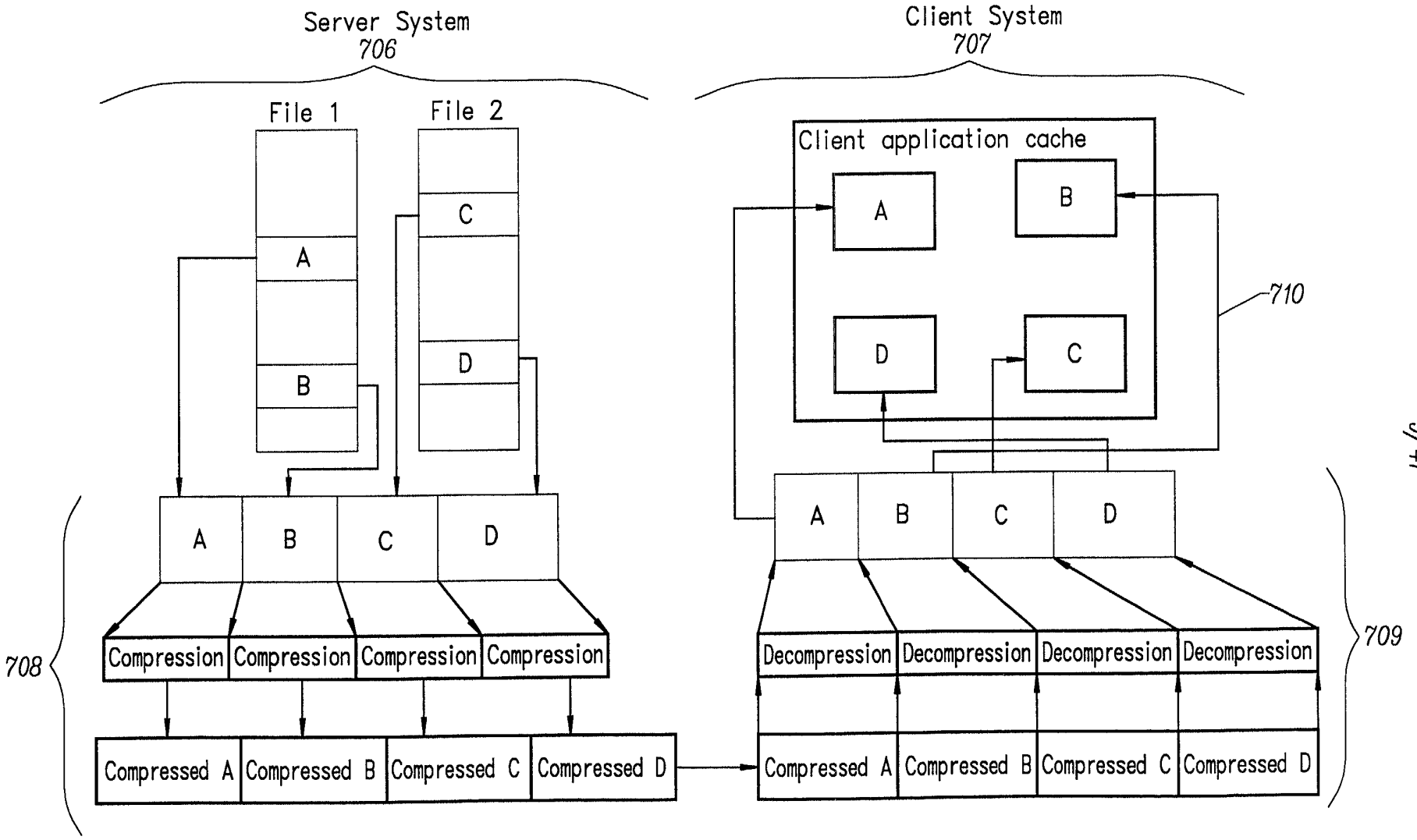


FIG. 7B

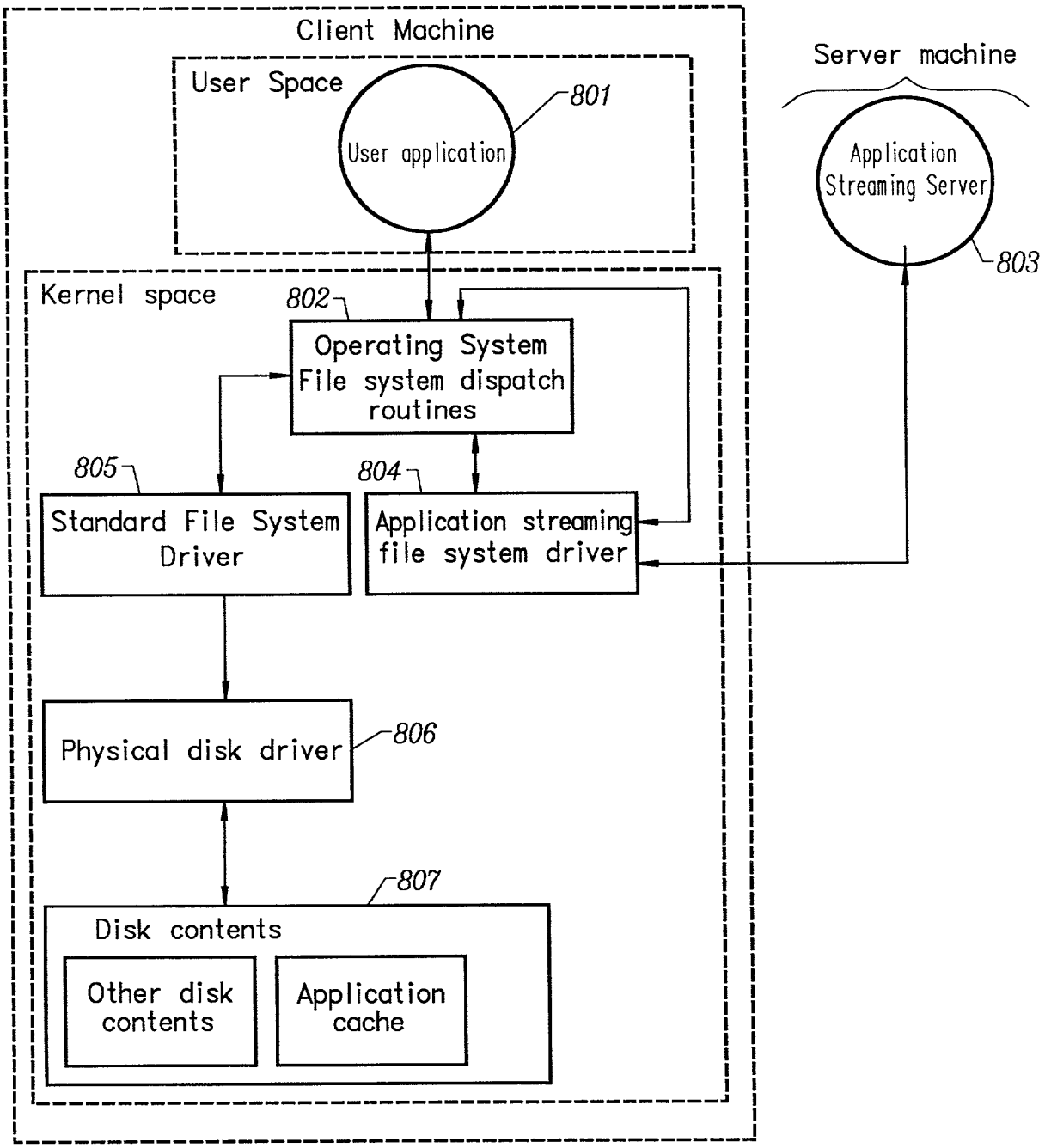

Client Machine
User Space
User application
801
Server machine
Application Streaming Server
803
Kernel space
802
Operating System File system dispatch routines
805
804
Standard File System Driver
Application streaming file system driver
Physical disk driver
806
807
Disk contents
Other disk contents
Application cache

FIG. 8

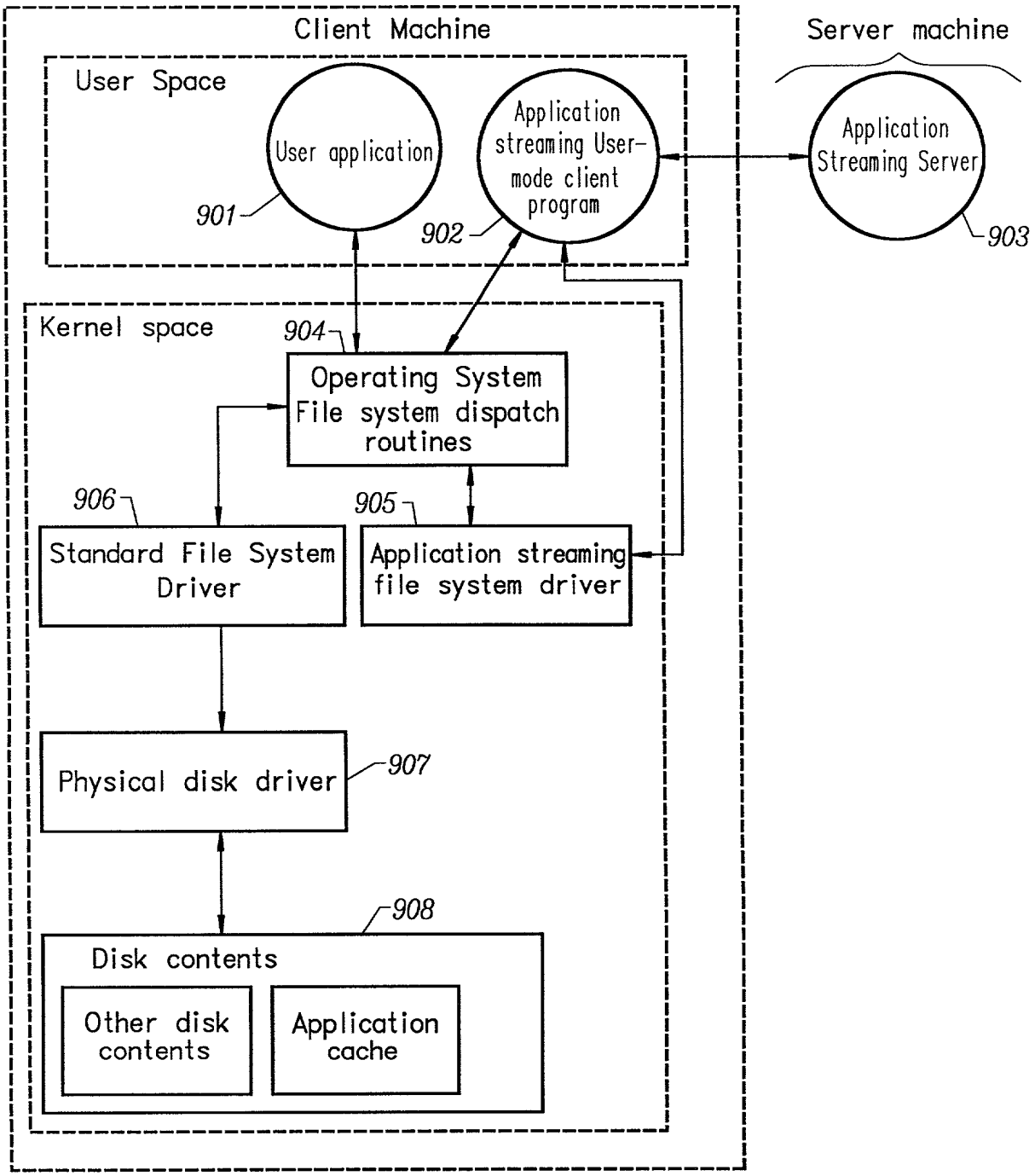

Client Machine
Server machine
User Space
User application
Application streaming User-mode client program
Application Streaming Server
901
902
903
Kernel space
904
Operating System File system dispatch routines
906
905
Standard File System Driver
Application streaming file system driver
Physical disk driver
907
908
Disk contents
Other disk contents
Application cache

FIG. 9

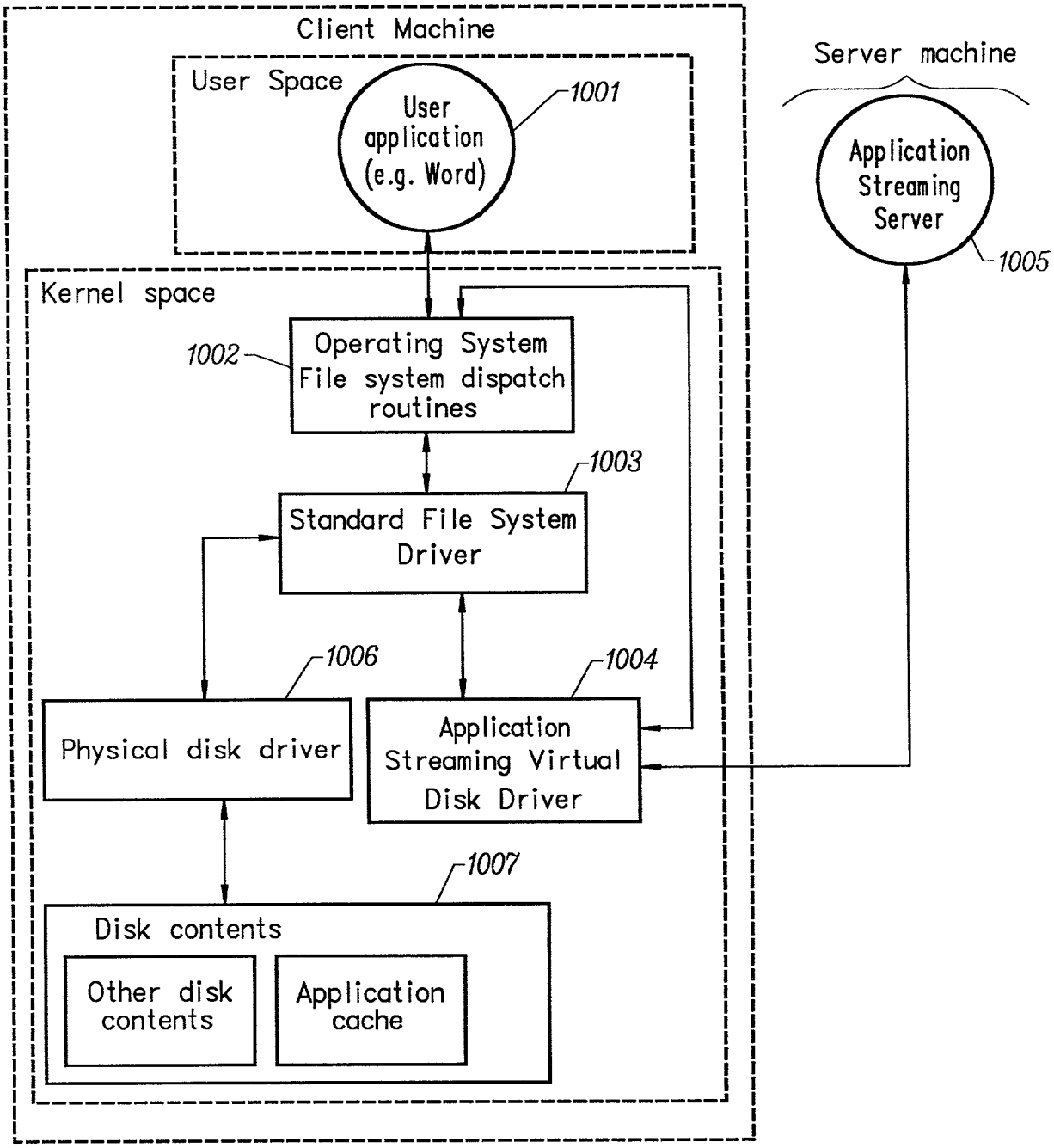

Client Machine
User Space
User application (e.g. Word)
1001
Server machine
Application Streaming Server
1005
Kernel space
1002
Operating System File system dispatch routines
1003
Standard File System Driver
1006
1004
Physical disk driver
Application Streaming Virtual Disk Driver
1007
Disk contents
Other disk contents
Application cache

FIG. 10

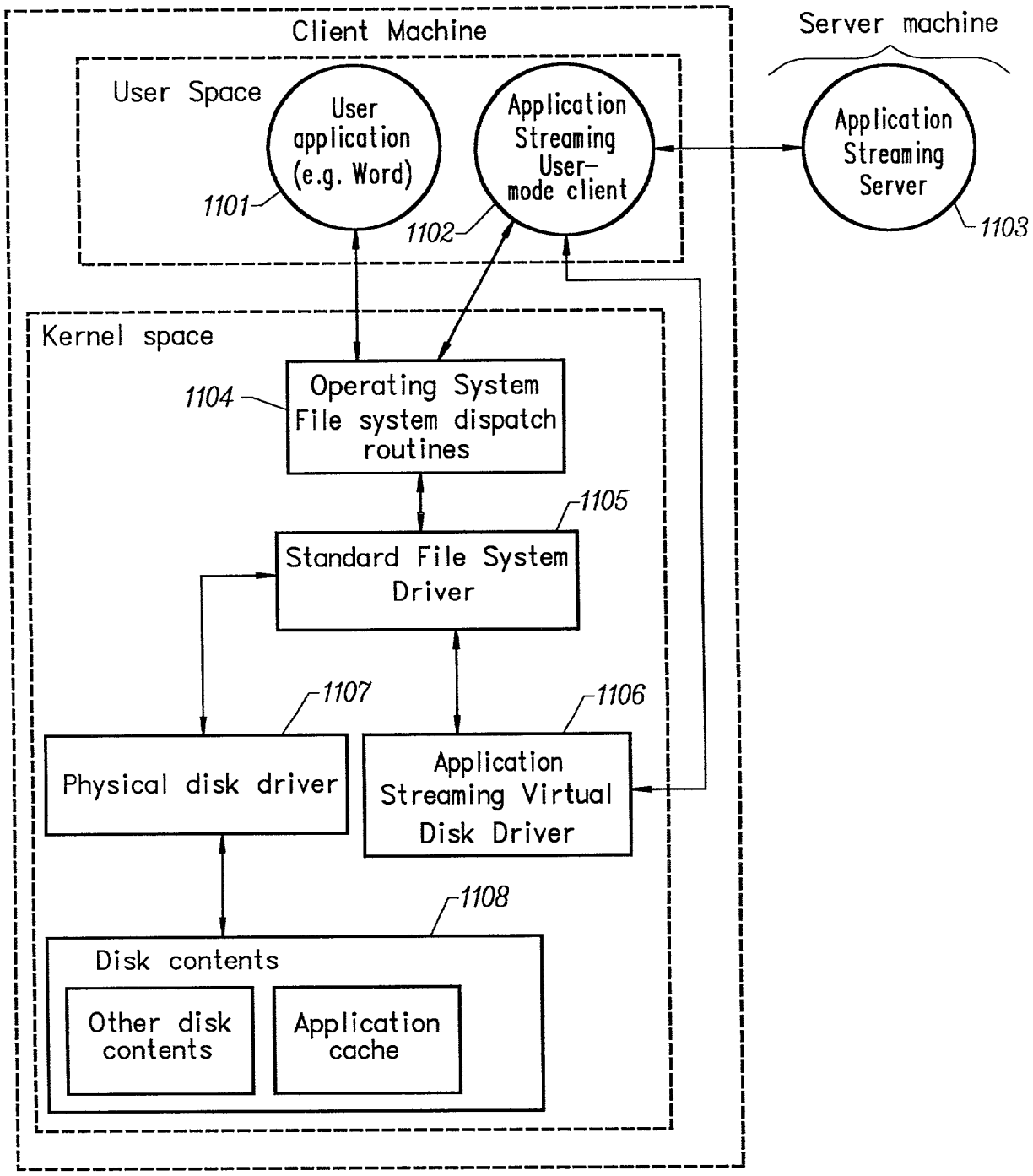

Client Machine
Server machine
User Space
User application (e.g. Word)
Application Streaming User-mode client
Application Streaming Server
1101
1102
1103
Kernel space
1104
Operating System File system dispatch routines
1105
Standard File System Driver
1107
1106
Physical disk driver
Application Streaming Virtual Disk Driver
1108
Disk contents
Other disk contents
Application cache

FIG. 11

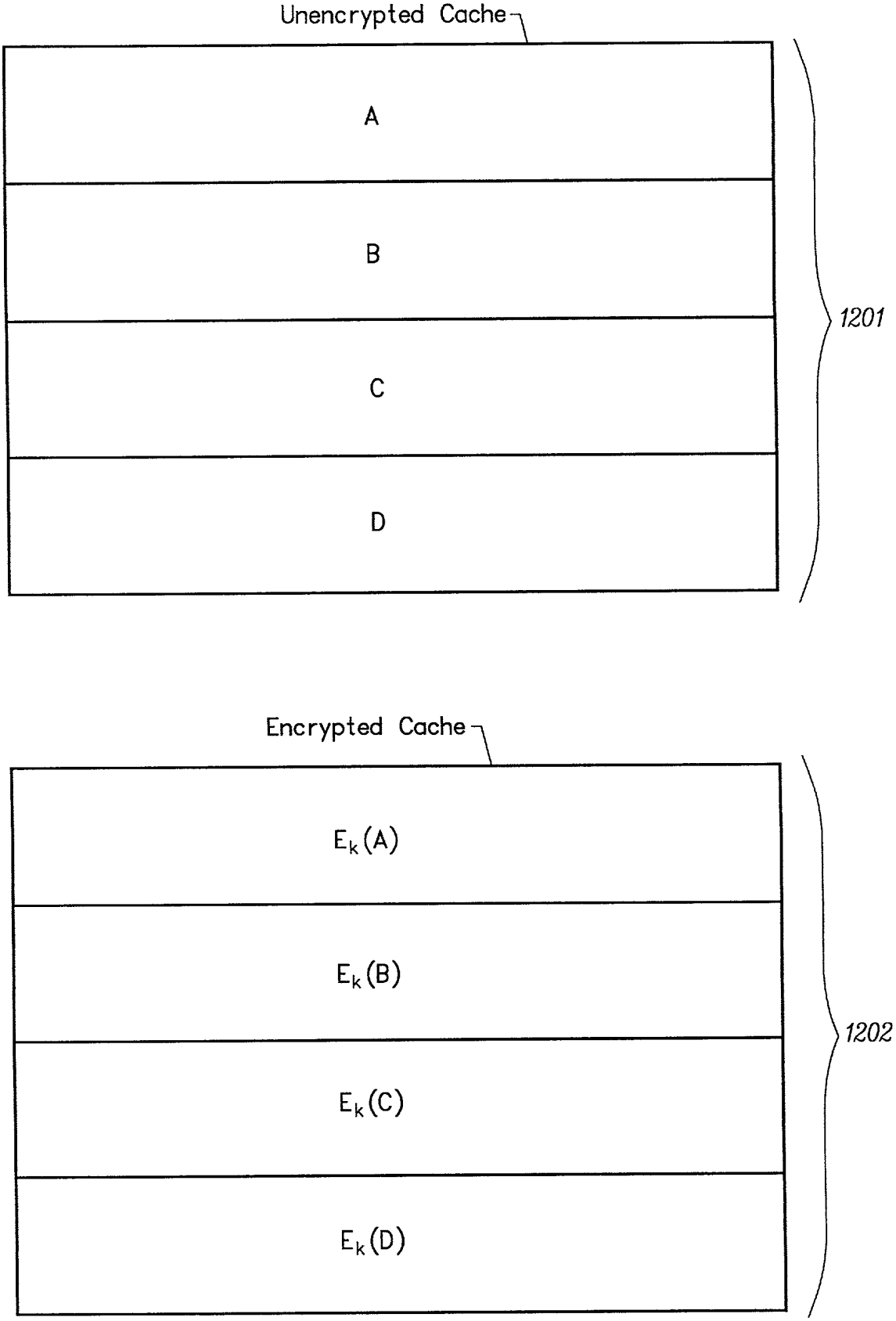

Unencrypted Cache
A
B
C
D
1201
Encrypted Cache
$E_k(A)$
$E_k(B)$
$E_k(C)$
$E_k(D)$
1202

FIG. 12

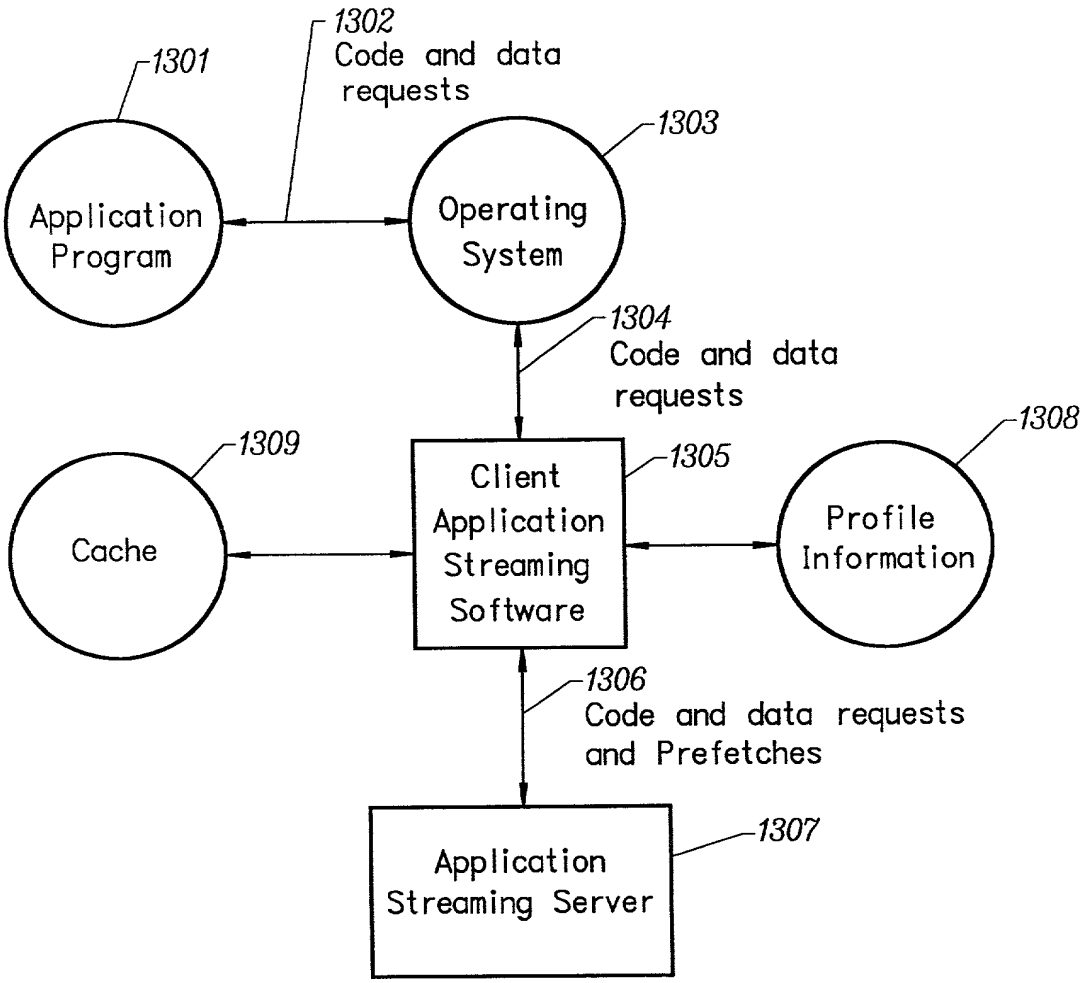

1301
1302
Code and data requests
1303
Application Program
Operating System
1304
Code and data requests
1308
1309
1305
Cache
Client Application Streaming Software
Profile Information
1306
Code and data requests and Prefetches
1307
Application Streaming Server

FIG. 13

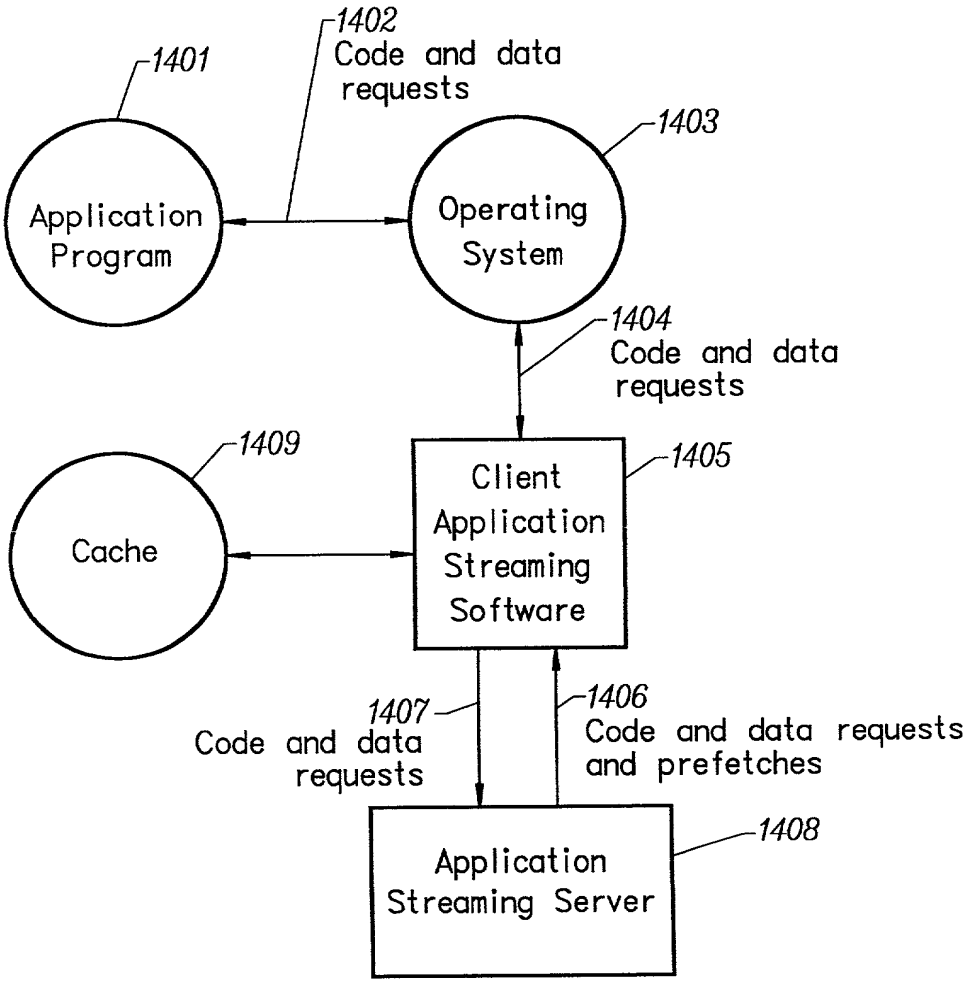

1401
1402
Code and data requests
1403
Application Program
Operating System
1404
Code and data requests
1409
1405
Cache
Client Application Streaming Software
1407
Code and data requests
1406
Code and data requests and prefetches
1408
Application Streaming Server

FIG. 14

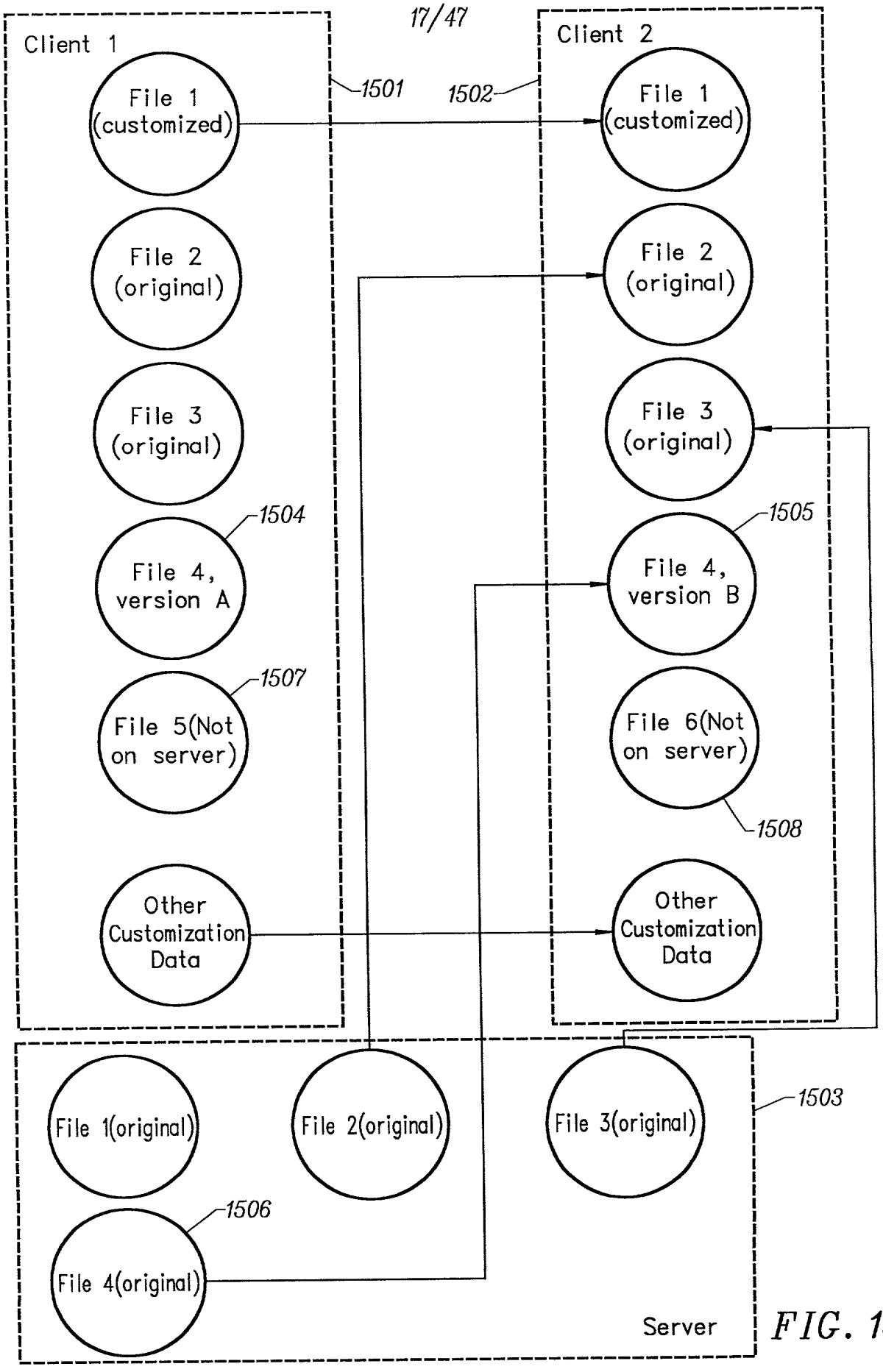

Client 1
File 1 (customized)
File 2 (original)
File 3 (original)
File 4, version A
File 5(Not on server)
Other Customization Data
1501
1504
1507
Client 2
File 1 (customized)
File 2 (original)
File 3 (original)
File 4, version B
File 6(Not on server)
Other Customization Data
1502
1505
1508
File 1(original)
File 2(original)
File 3(original)
File 4(original)
Server
1503
1506

FIG. 15

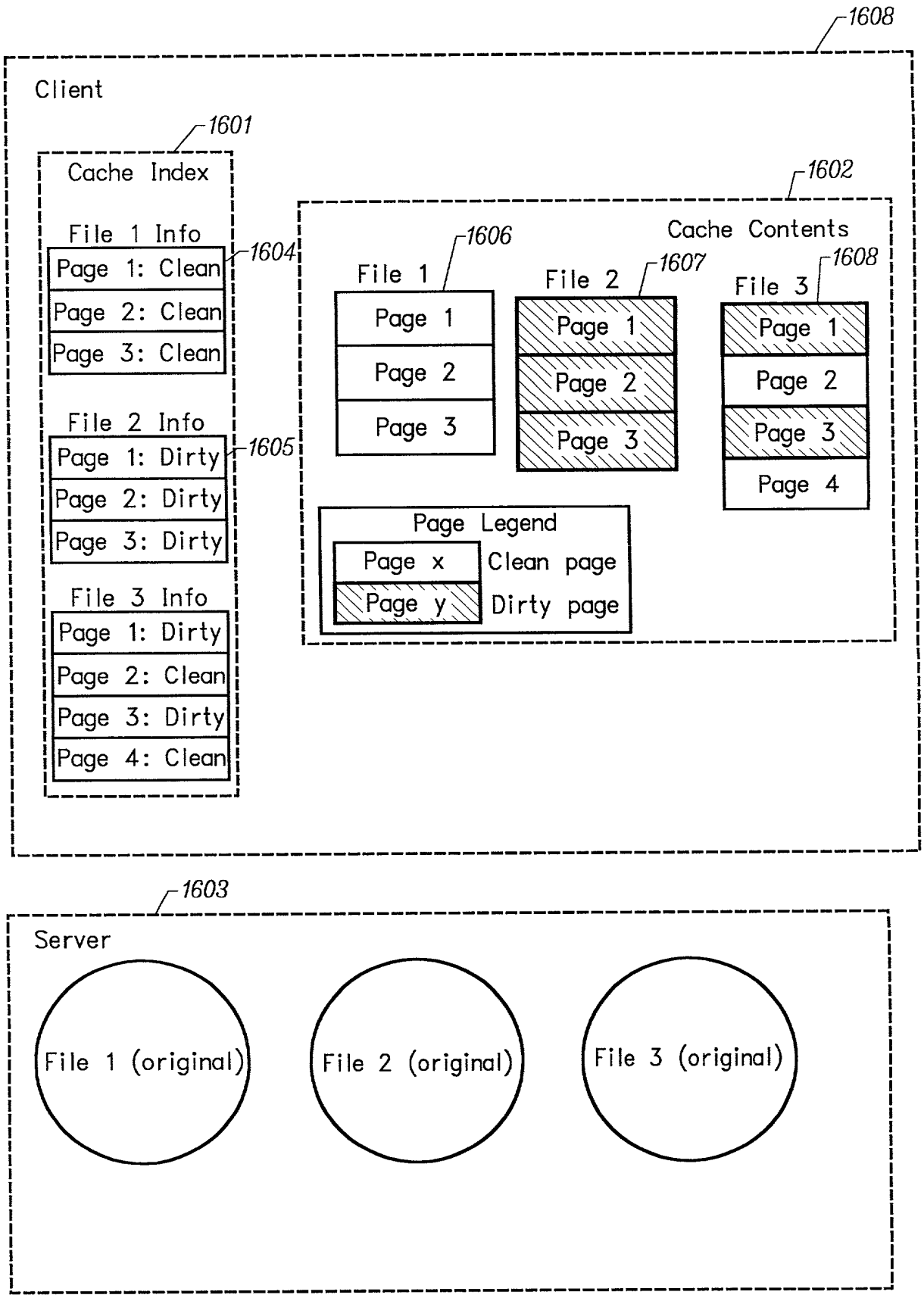

1608
Client
1601
Cache Index
File 1 Info
Page 1: Clean
Page 2: Clean
Page 3: Clean
1604
File 2 Info
Page 1: Dirty
Page 2: Dirty
Page 3: Dirty
1605
File 3 Info
Page 1: Dirty
Page 2: Clean
Page 3: Dirty
Page 4: Clean
1602
Cache Contents
1606
File 1
Page 1
Page 2
Page 3
1607
File 2
Page 1
Page 2
Page 3
1608
File 3
Page 1
Page 2
Page 3
Page 4
Page Legend
Page x
Clean page
Page y
Dirty page
1603
Server
File 1 (original)
File 2 (original)
File 3 (original)

FIG. 16

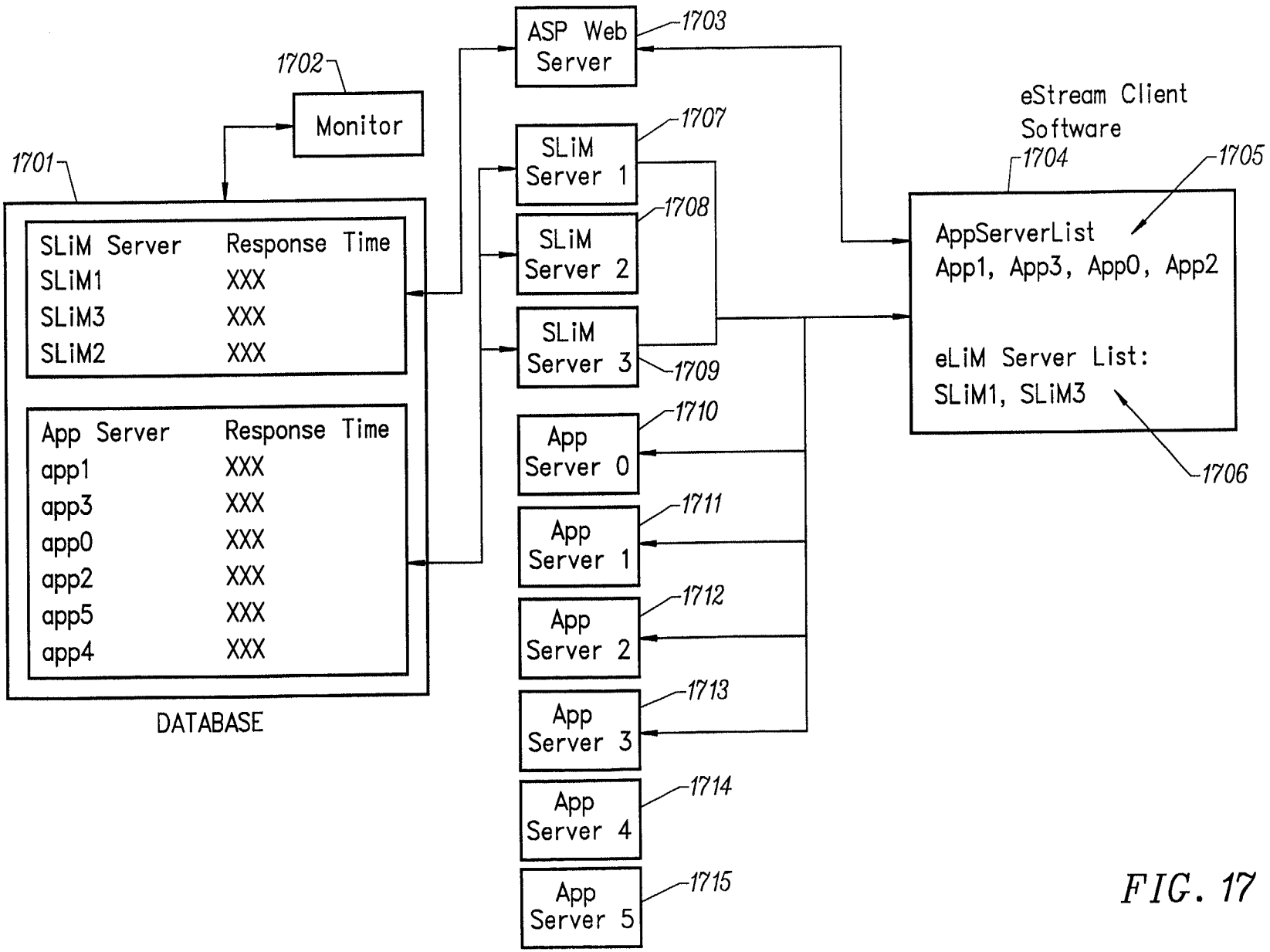
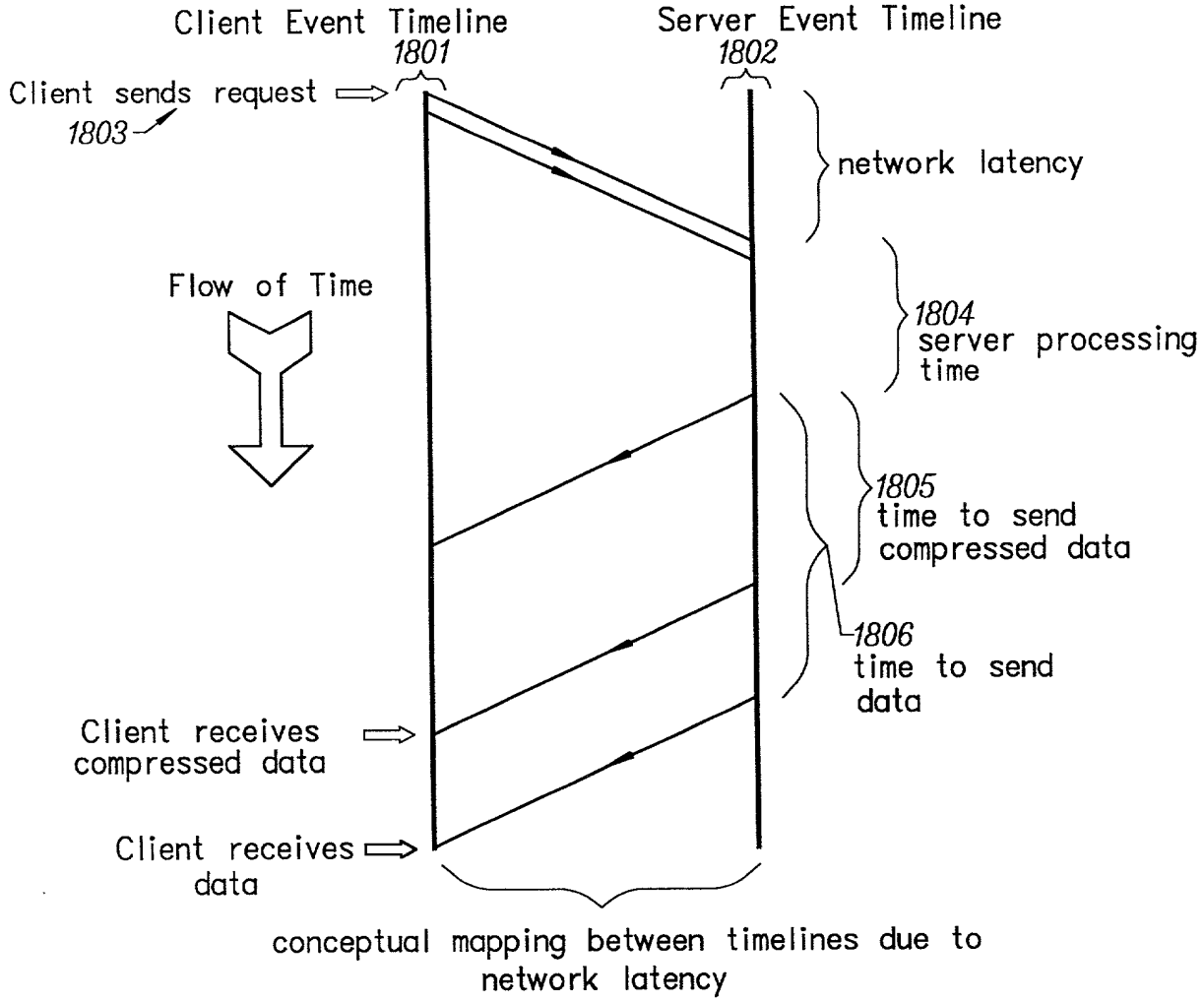

ASP Web Server
1703
1702
Monitor
1701
SLiM Server 1
1707
SLiM Server 2
1708
SLiM Server 3
1709
eStream Client Software
1704
1705
AppServerList
App1, App3, App0, App2
eLiM Server List:
SLiM1, SLiM3
1706
SLiM Server | Response Time
SLiM1 | XXX
SLiM3 | XXX
SLiM2 | XXX
App Server | Response Time
app1 | XXX
app3 | XXX
app0 | XXX
app2 | XXX
app5 | XXX
app4 | XXX
DATABASE
App Server 0
1710
App Server 1
1711
App Server 2
1712
App Server 3
1713
App Server 4
1714
App Server 5
1715

FIG. 17



Client receives data sooner if it is compressed

FIG. 18

PCT/US00/26850

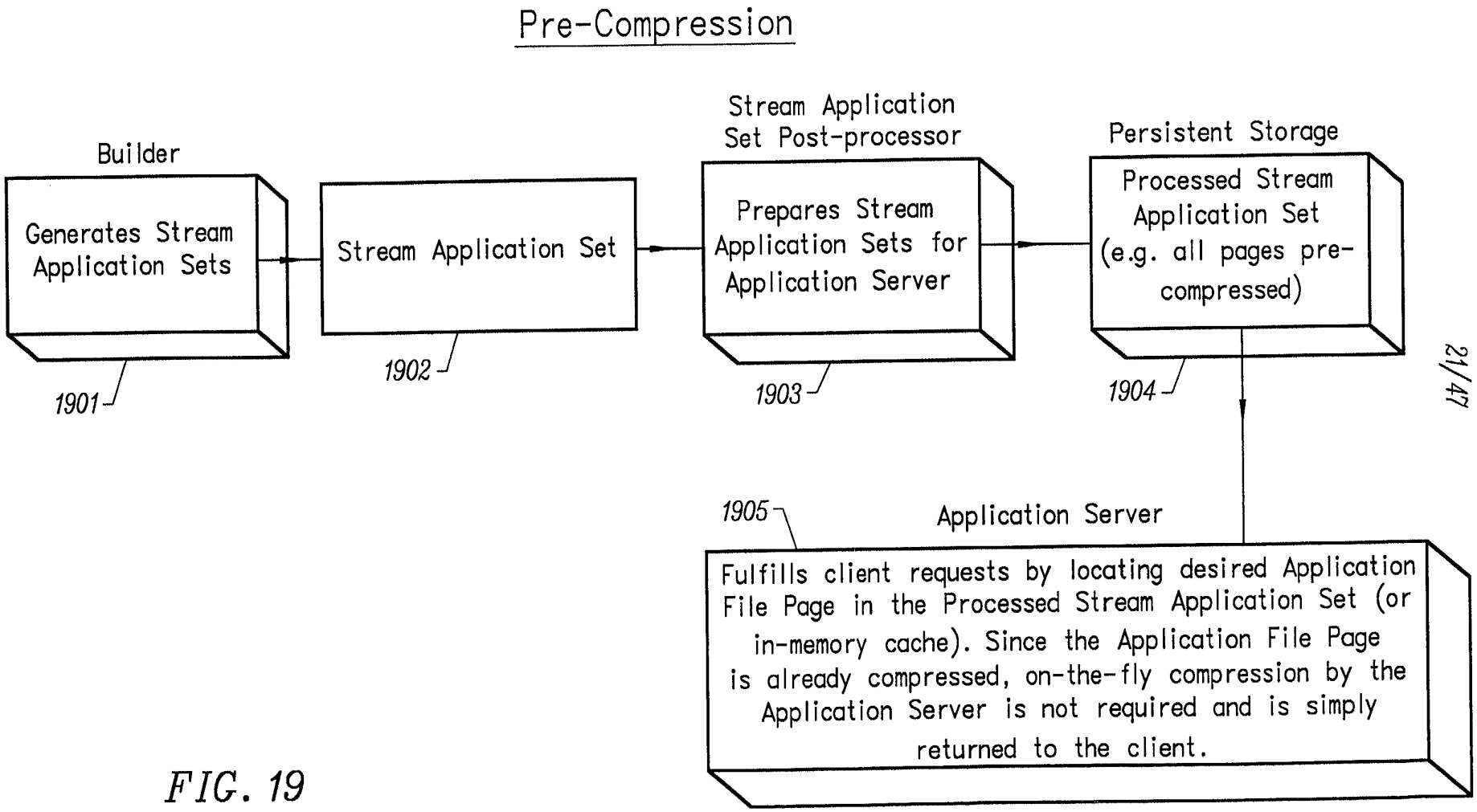
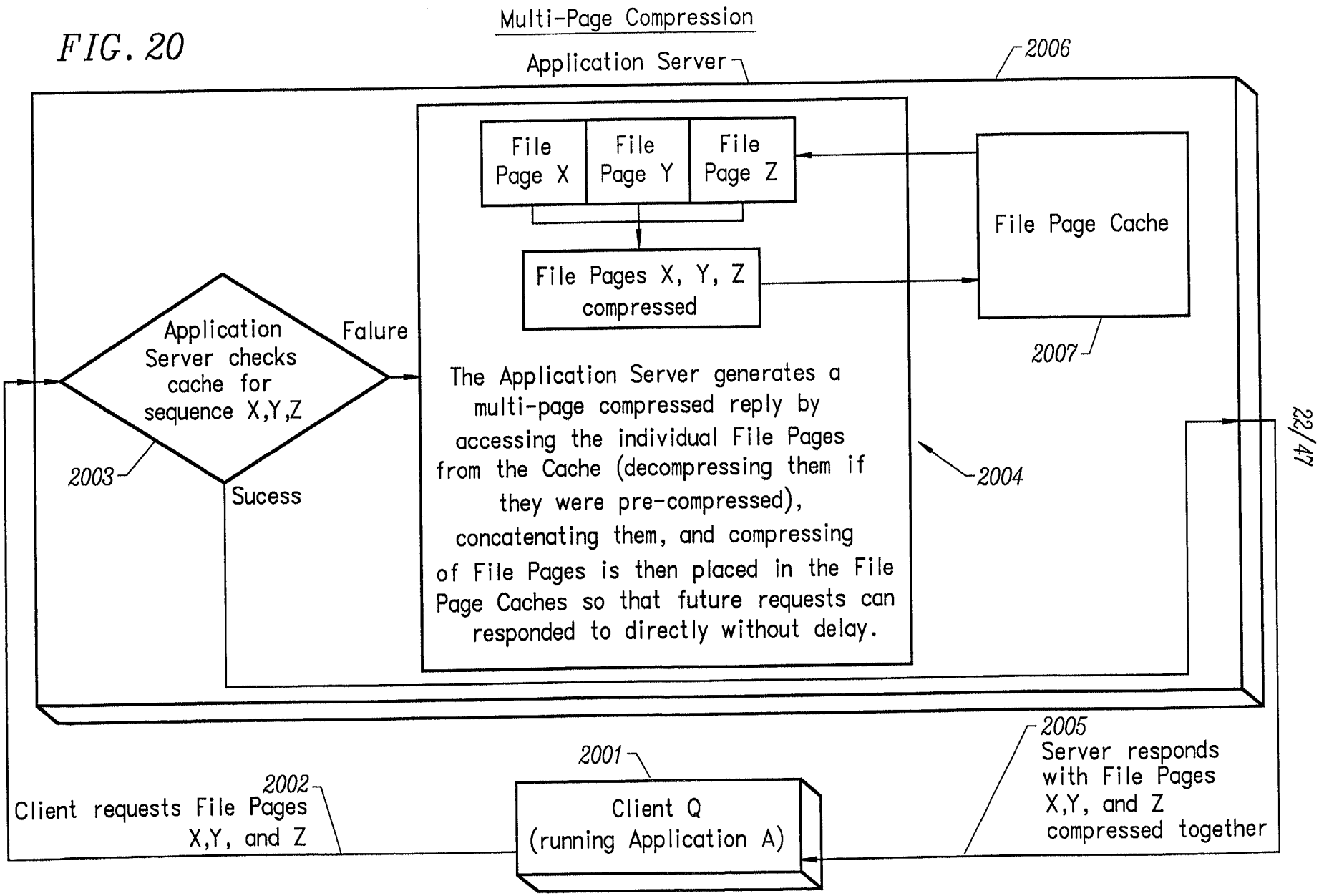


FIG. 19

FIG. 20


Multi-Page Compression
Application Server
2006
File Page X
File Page Y
File Page Z
File Page Cache
File Pages X, Y, Z compressed
2007
Application Server checks cache for sequence X,Y,Z
Falure
Sucess
2003
The Application Server generates a multi-page compressed reply by accessing the individual File Pages from the Cache (decompressing them if they were pre-compressed), concatenating them, and compressing of File Pages is then placed in the File Page Caches so that future requests can responded to directly without delay.
2004
2001
Client Q (running Application A)
2002
Client requests File Pages X,Y, and Z
2005
Server responds with File Pages X,Y, and Z compressed together

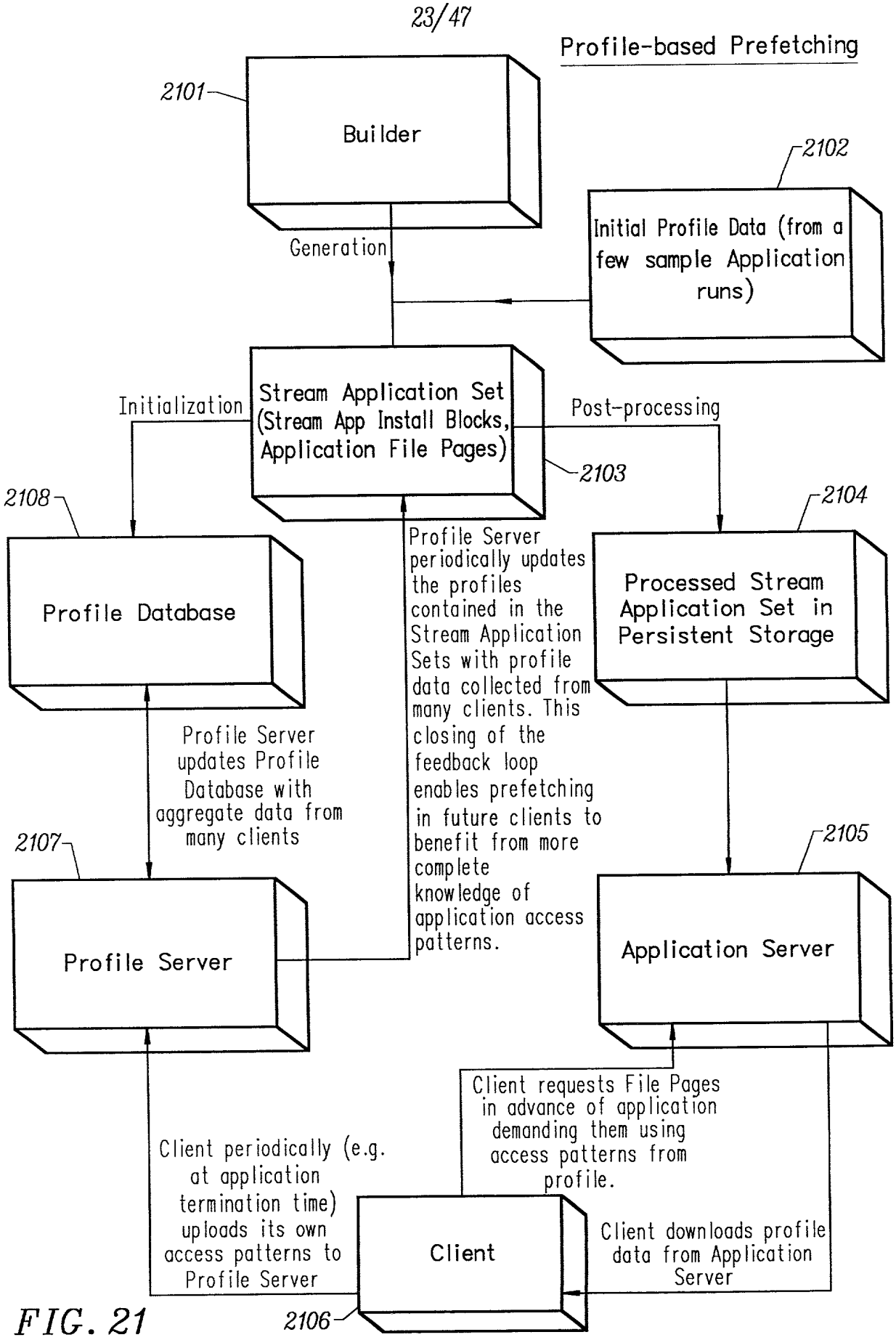

Profile-based Prefetching
2101
Builder
2102
Initial Profile Data (from a few sample Application runs)
Generation
Stream Application Set (Stream App Install Blocks, Application File Pages)
2103
Initialization
Post-processing
2108
Profile Database
2104
Processed Stream Application Set in Persistent Storage
Profile Server periodically updates the profiles contained in the Stream Application Sets with profile data collected from many clients. This closing of the feedback loop enables prefetching in future clients to benefit from more complete knowledge of application access patterns.
Profile Server updates Profile Database with aggregate data from many clients
2107
Profile Server
2105
Application Server
Client requests File Pages in advance of application demanding them using access patterns from profile.
Client periodically (e.g. at application termination time) uploads its own access patterns to Profile Server
Client
Client downloads profile data from Application Server
2106

FIG. 21

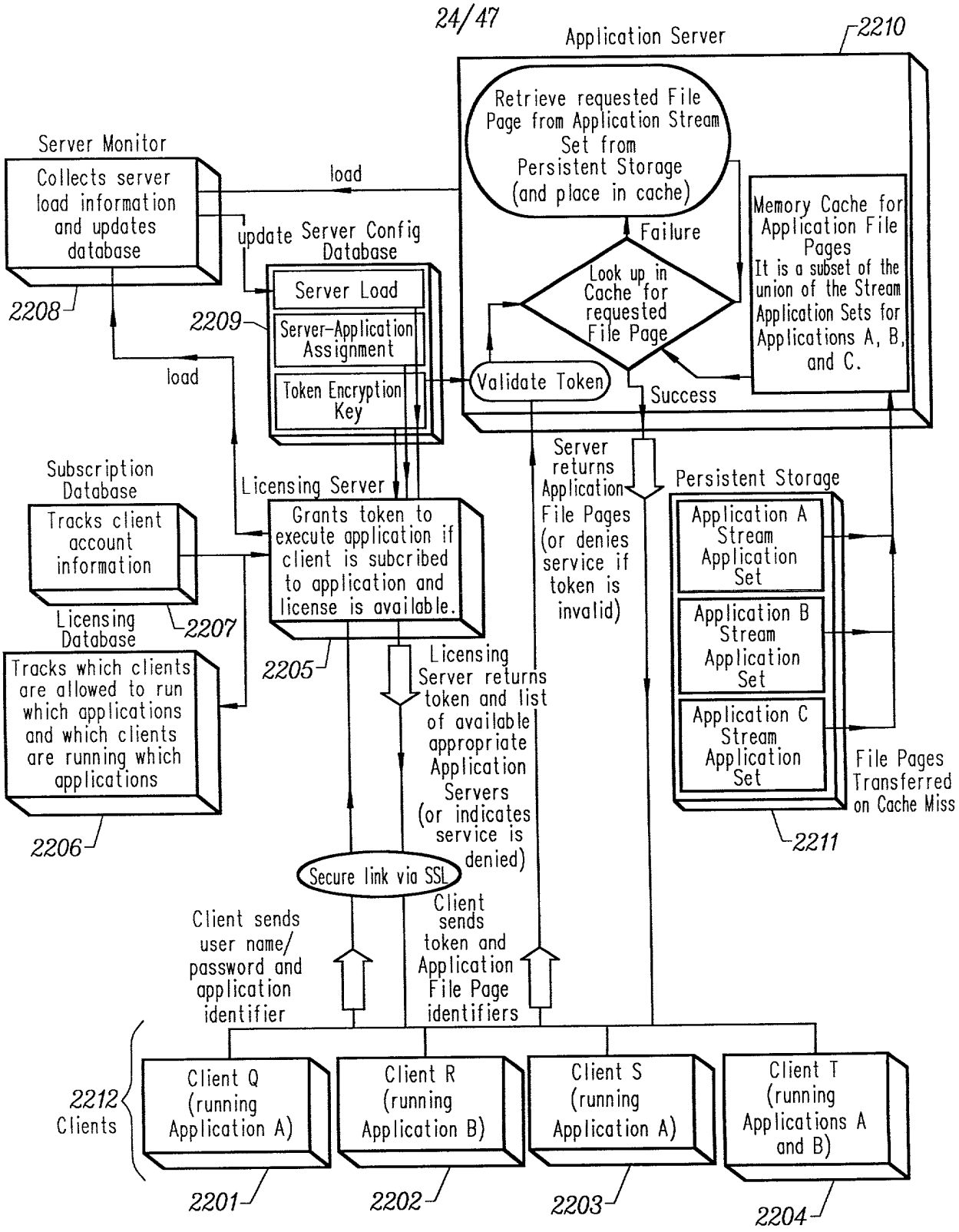

Application Server
2210
Retrieve requested File Page from Application Stream Set from Persistent Storage (and place in cache)
Memory Cache for Application File Pages
It is a subset of the union of the Stream Application Sets for Applications A, B, and C.
Failure
Look up in Cache for requested File Page
Validate Token
Success
Server Monitor
Collects server load information and updates database
load
update Server Config Database
Server Load
Server-Application Assignment
Token Encryption Key
2208
2209
load
Subscription Database
Tracks client account information
2207
Licensing Server
Grants token to execute application if client is subcribed to application and license is available.
2205
Server returns Application File Pages (or denies service if token is invalid)
Persistent Storage
Application A Stream Application Set
Application B Stream Application Set
Application C Stream Application Set
File Pages Transferred on Cache Miss
2211
Licensing Database
Tracks which clients are allowed to run which applications and which clients are running which applications
2206
Licensing Server returns token and list of available appropriate Application Servers (or indicates service is denied)
Secure link via SSL
Client sends user name/ password and application identifier
Client sends token and Application File Page identifiers
2212 Clients
Client Q (running Application A)
Client R (running Application B)
Client S (running Application A)
Client T (running Applications A and B)
2201
2202
2203
2204

FIG. 22

Builder Install Monitor (IM) Control Flow Diagram

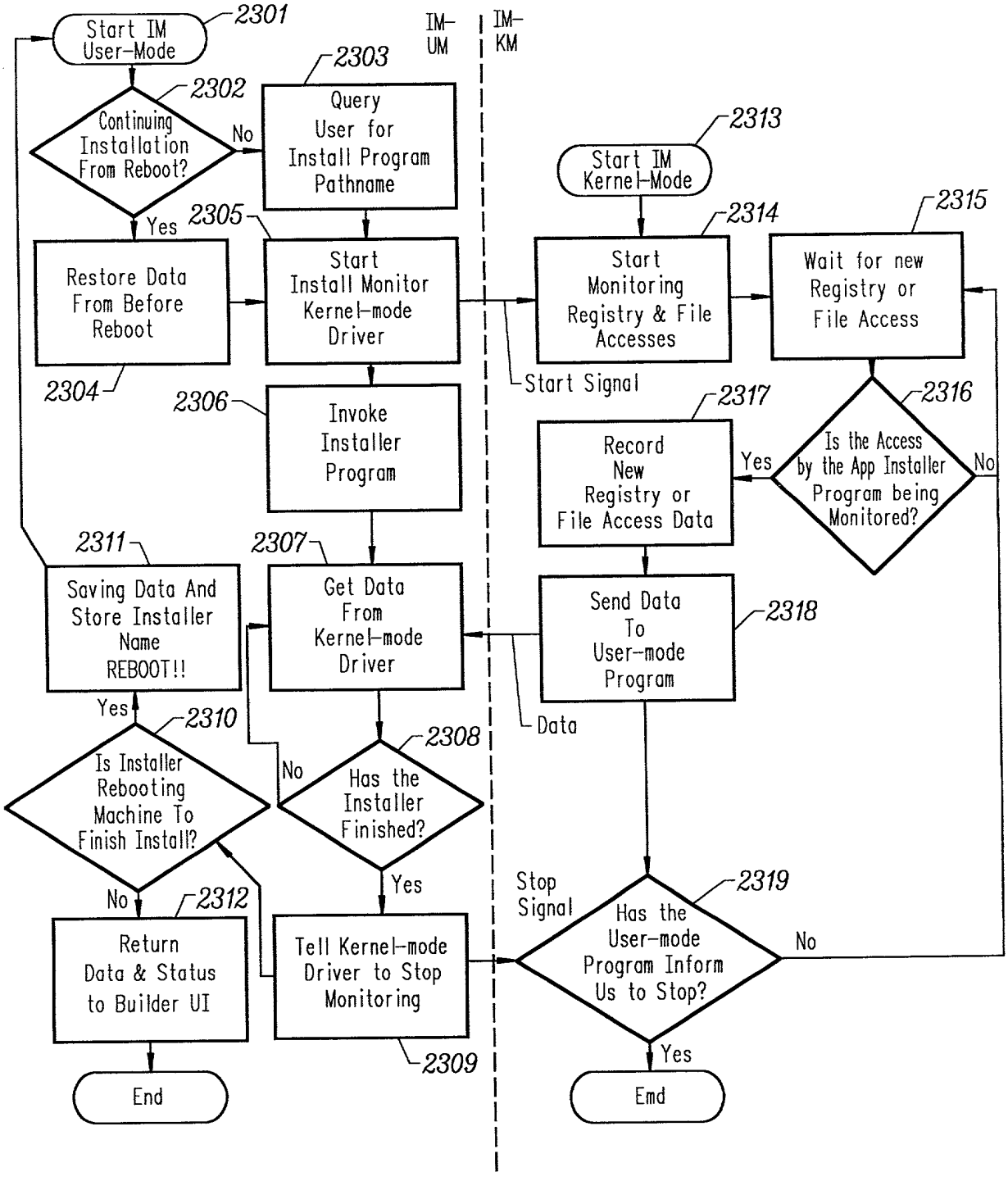


FIG. 23

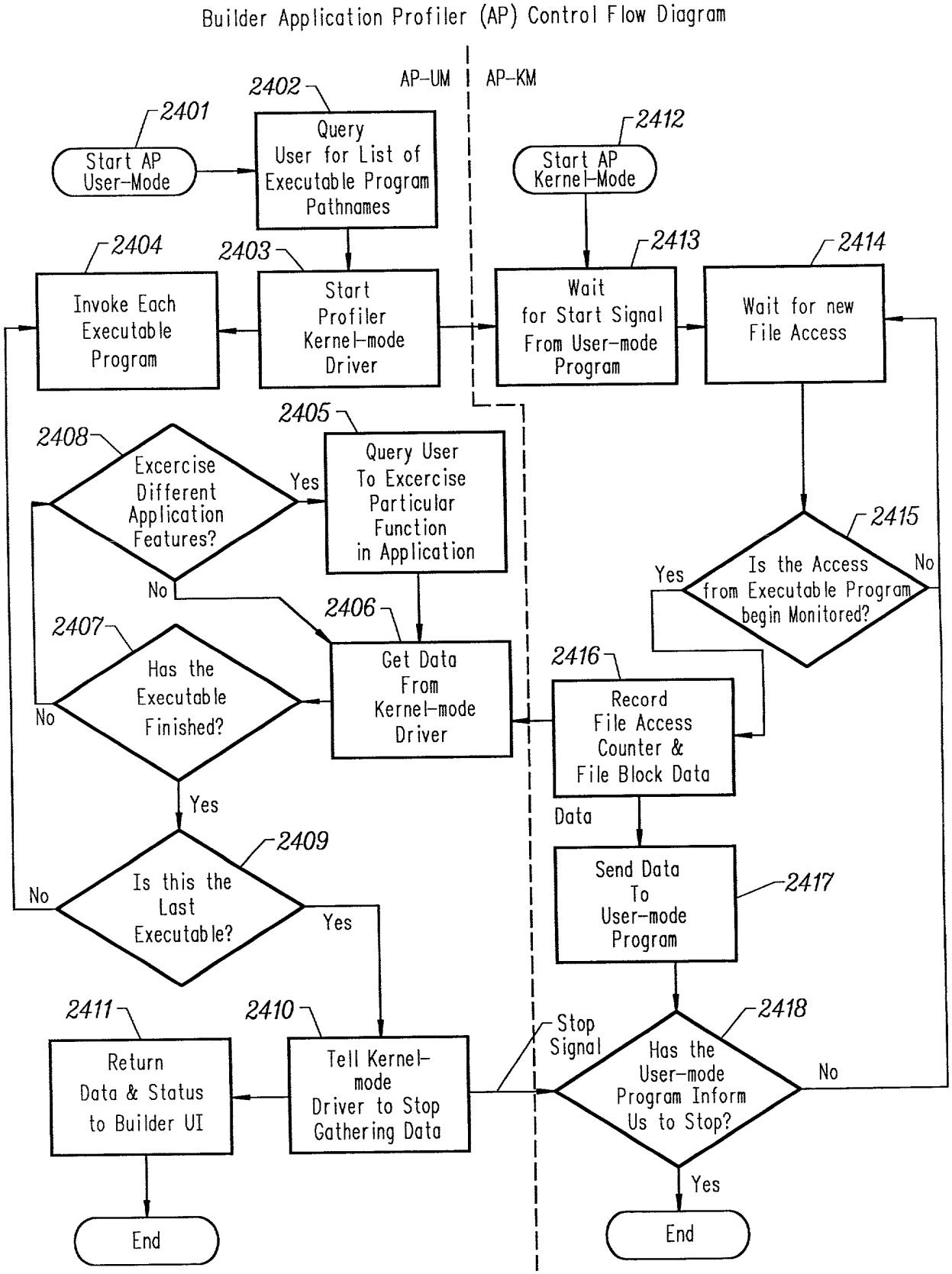

Builder Application Profiler (AP) Control Flow Diagram
AP-UM
AP-KM
2401
Start AP User-Mode
2402
Query User for List of Executable Program Pathnames
2403
Start Profiler Kernel-mode Driver
2404
Invoke Each Executable Program
2405
Query User To Excercise Particular Function in Application
2406
Get Data From Kernel-mode Driver
2407
Has the Executable Finished?
2408
Excercise Different Application Features?
2409
Is this the Last Executable?
2410
Tell Kernel-mode Driver to Stop Gathering Data
2411
Return Data & Status to Builder UI
End
2412
Start AP Kernel-Mode
2413
Wait for Start Signal From User-mode Program
2414
Wait for new File Access
2415
Is the Access from Executable Program begin Monitored?
2416
Record File Access Counter & File Block Data
Data
2417
Send Data To User-mode Program
2418
Has the User-mode Program Inform Us to Stop?
Stop Signal
End
Yes
No

FIG. 24

Builder SAS Packager (SP) Control Flow Diagram

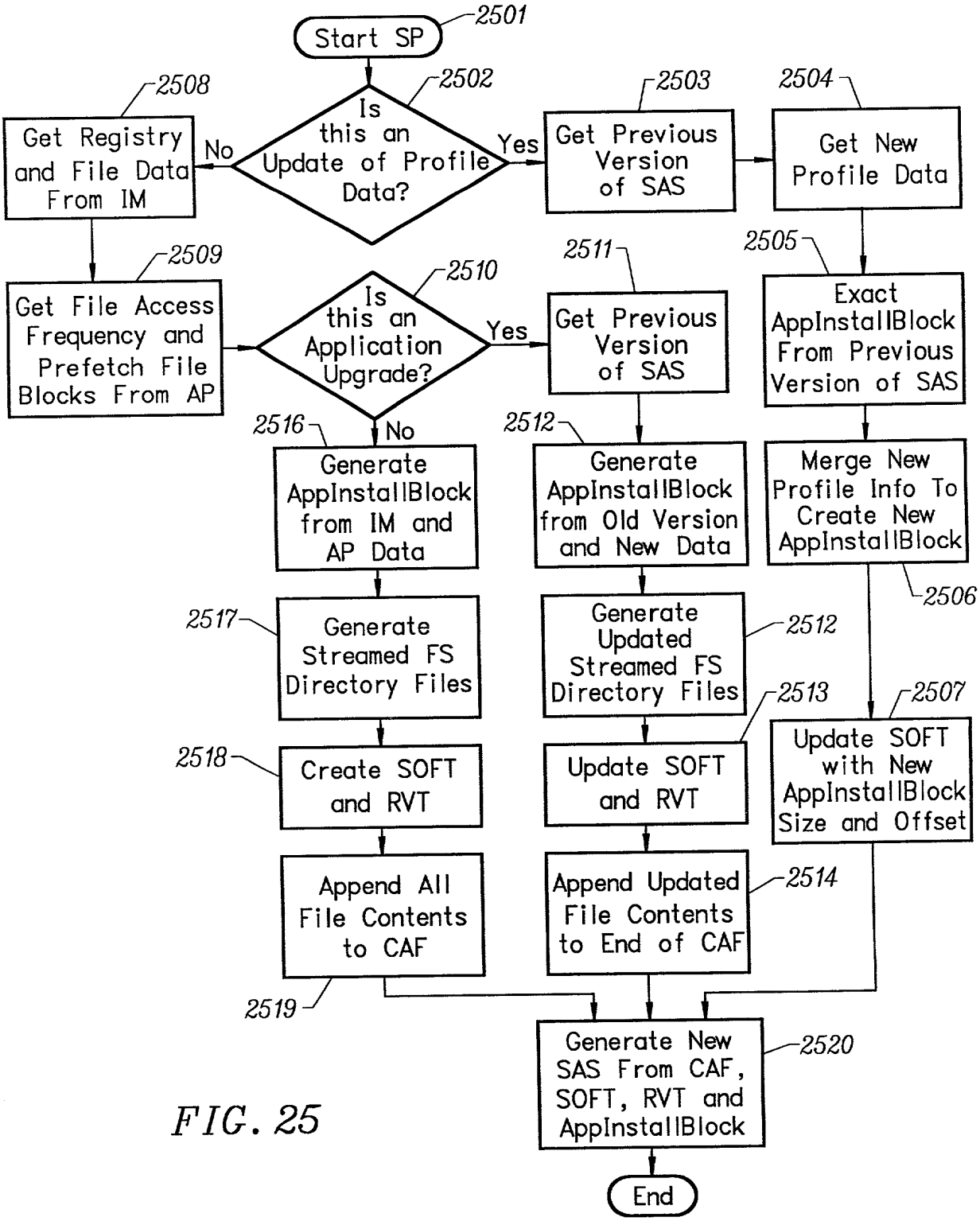


FIG. 25

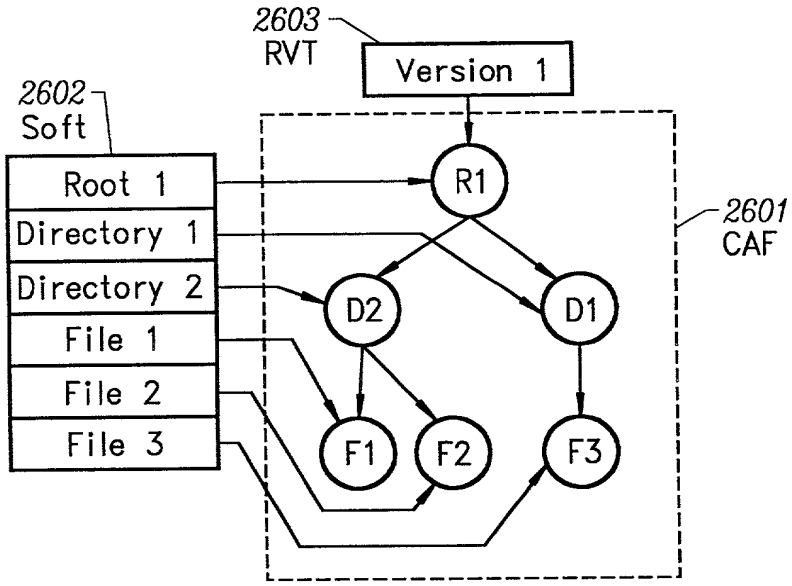


FIG. 26A

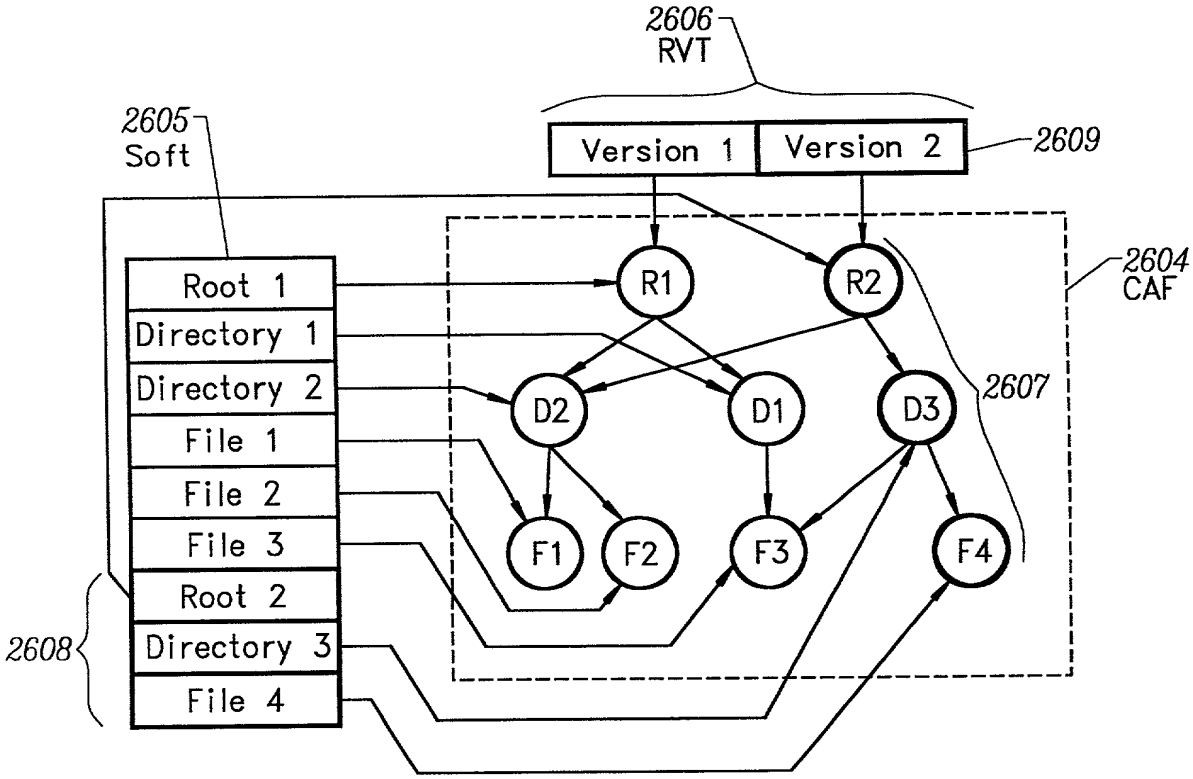


FIG. 26B

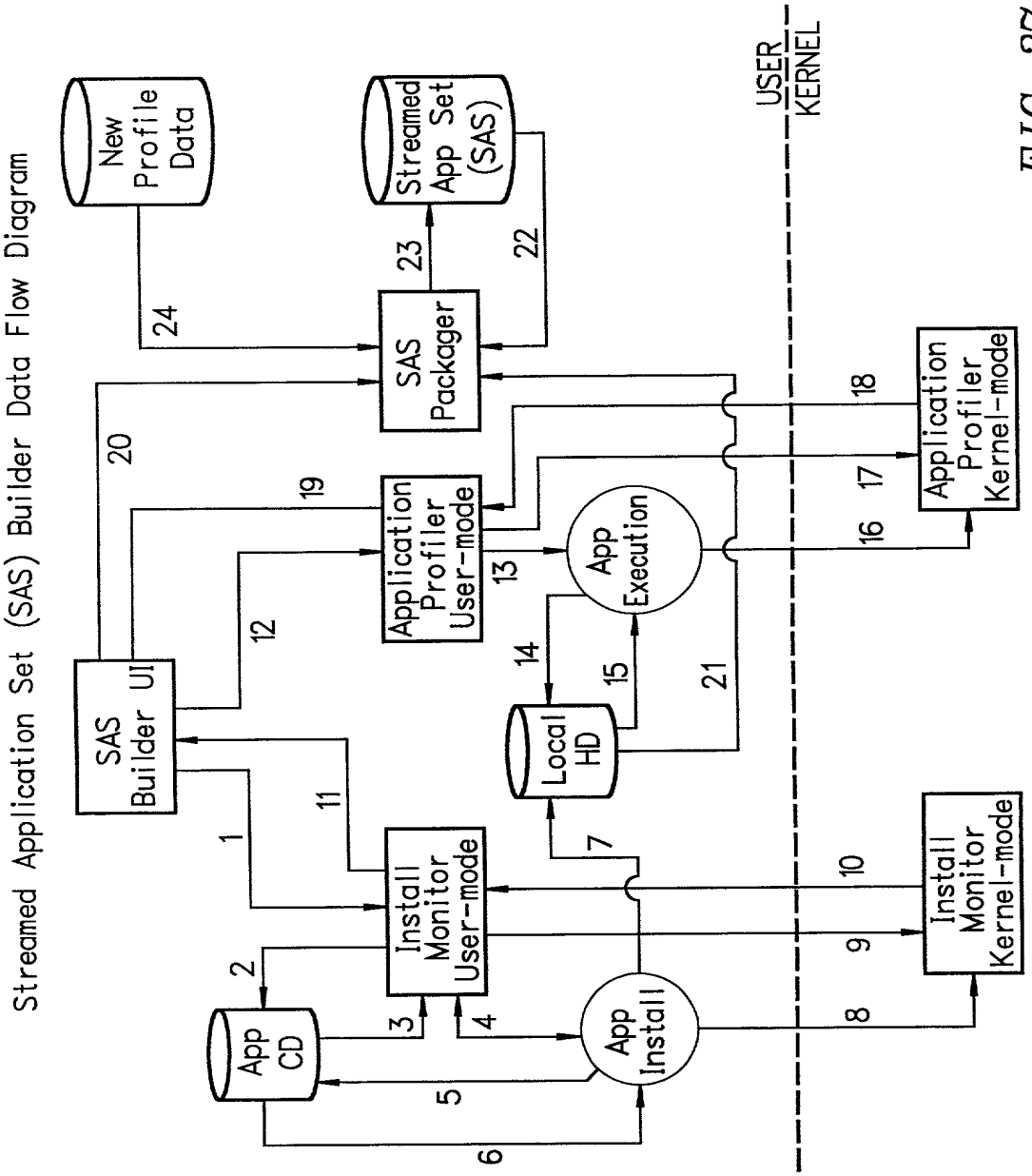

Streamed Application Set (SAS) Builder Data Flow Diagram
New Profile Data
Streamed App Set (SAS)
SAS Packager
SAS Builder UI
Application Profiler User-mode
App Execution
Application Profiler Kernel-mode
Local HD
Install Monitor User-mode
Install Monitor Kernel-mode
App CD
App Install
USER
KERNEL
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24

FIG. 27

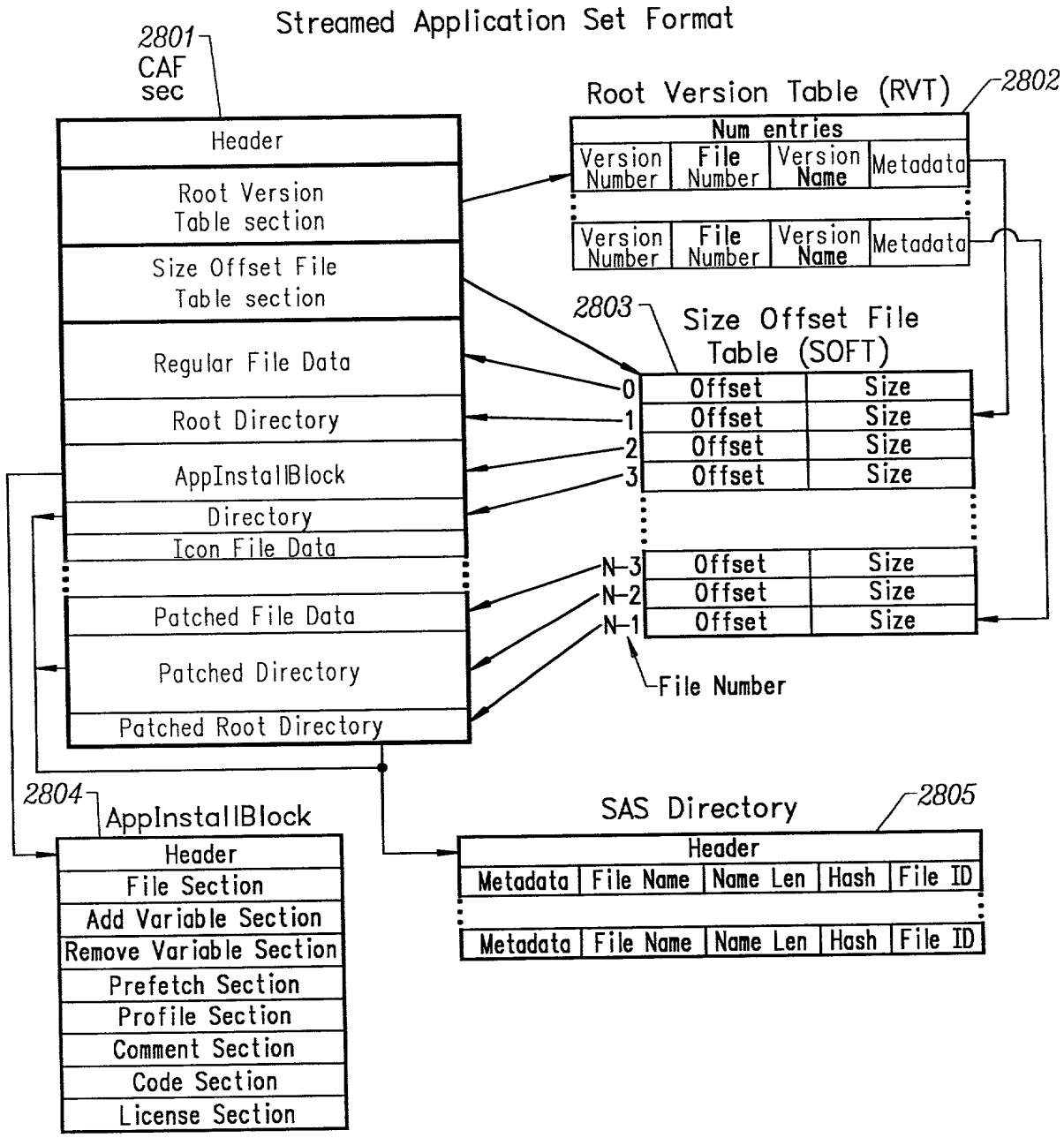

Streamed Application Set Format
2801
CAF
sec
Header
Root Version
Table section
Size Offset File
Table section
Regular File Data
Root Directory
AppInstallBlock
Directory
Icon File Data
Patched File Data
Patched Directory
Patched Root Directory
Root Version Table (RVT)
2802
Num entries
Version Number
File Number
Version Name
Metadata
2803
Size Offset File
Table (SOFT)
Offset
Size
0
1
2
3
N-3
N-2
N-1
File Number
2804
AppInstallBlock
Header
File Section
Add Variable Section
Remove Variable Section
Prefetch Section
Profile Section
Comment Section
Code Section
License Section
SAS Directory
2805
Header
Metadata
File Name
Name Len
Hash
File ID

FIG. 28

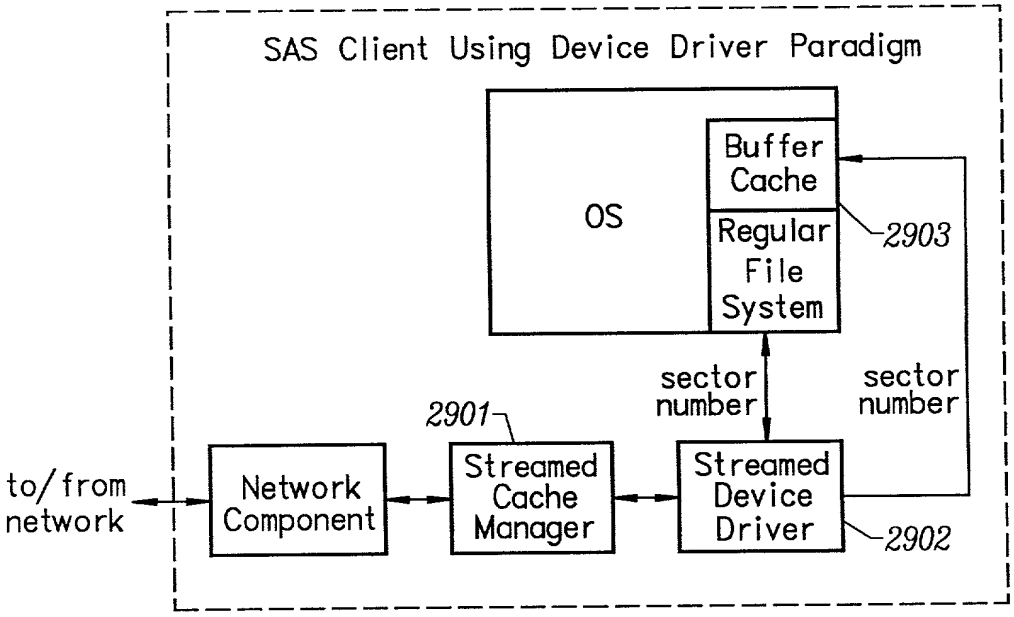

SAS Client Using Device Driver Paradigm
OS
Buffer Cache
Regular File System
2903
sector number
sector number
2901
to/from network
Network Component
Streamed Cache Manager
Streamed Device Driver
2902

FIG. 29

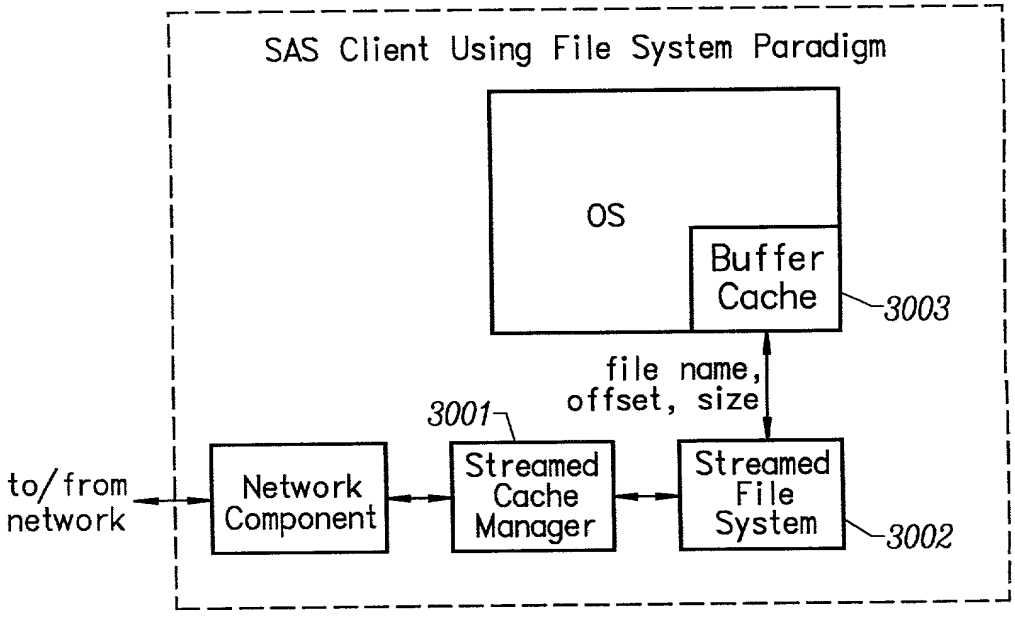

SAS Client Using File System Paradigm
OS
Buffer Cache
3003
file name, offset, size
3001
to/from network
Network Component
Streamed Cache Manager
Streamed File System
3002

FIG. 30

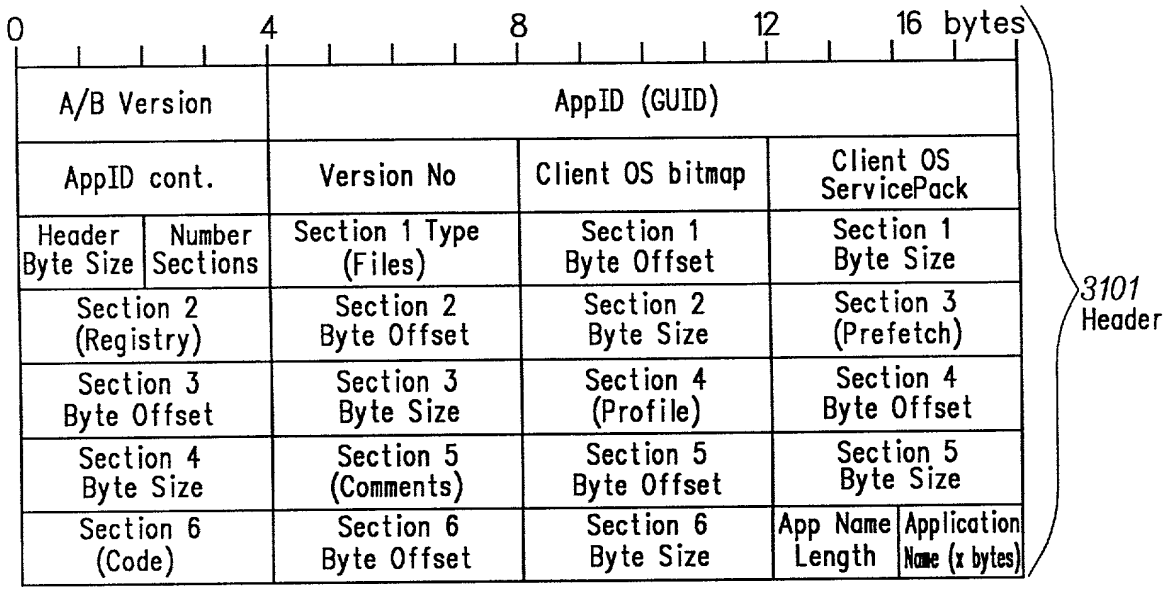


FIG. 31A

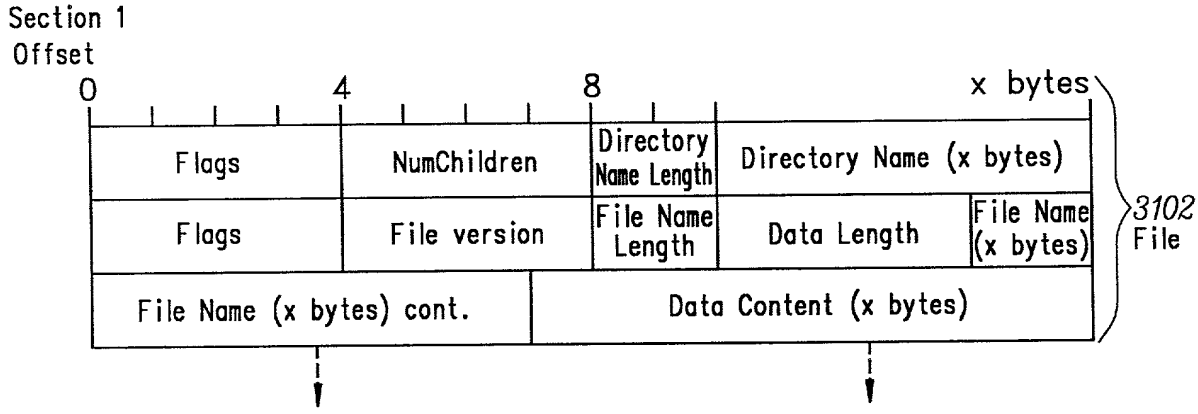


FIG. 31B

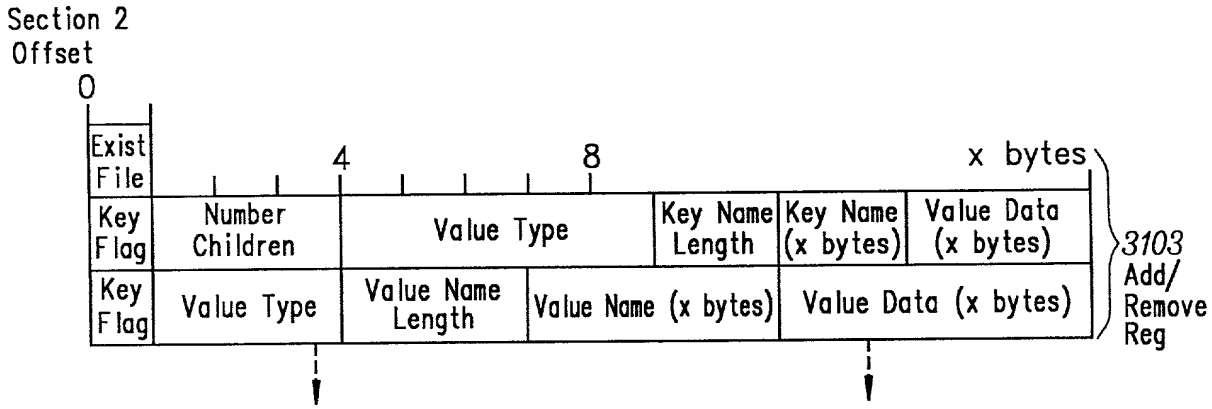


FIG. 31C

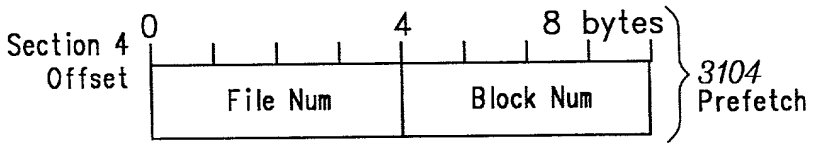


FIG. 31D

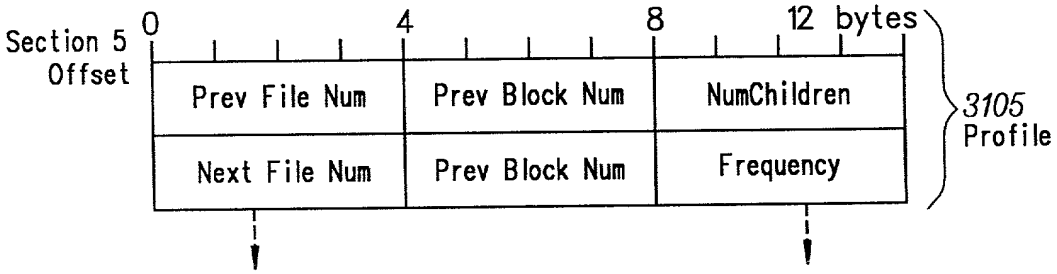


FIG. 31E

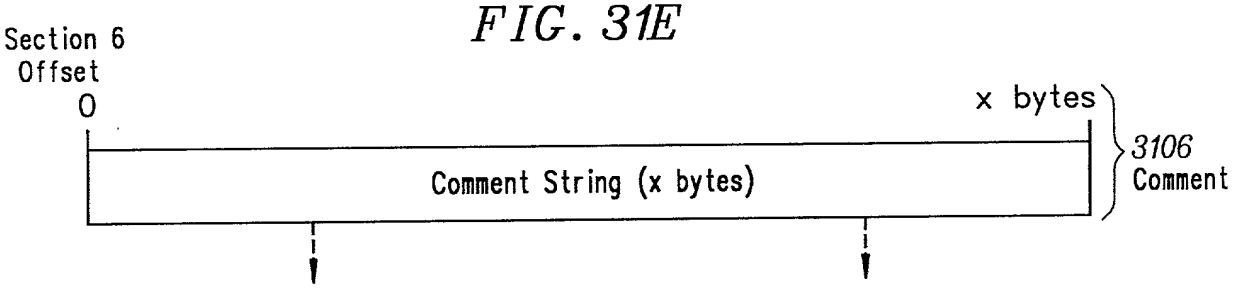


FIG. 31F

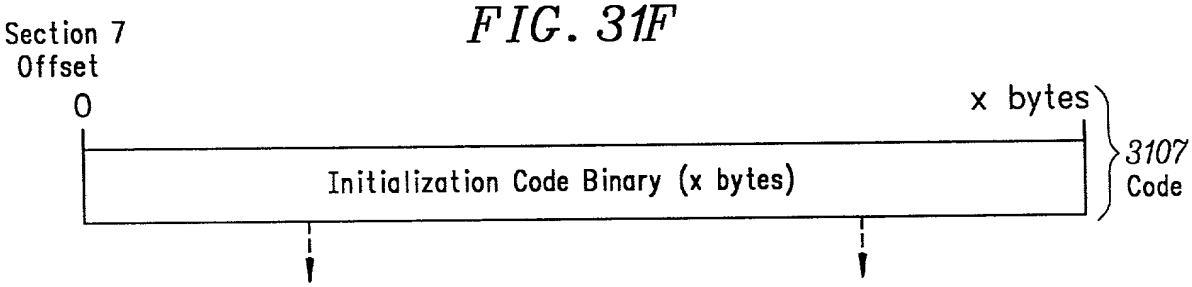


FIG. 31G

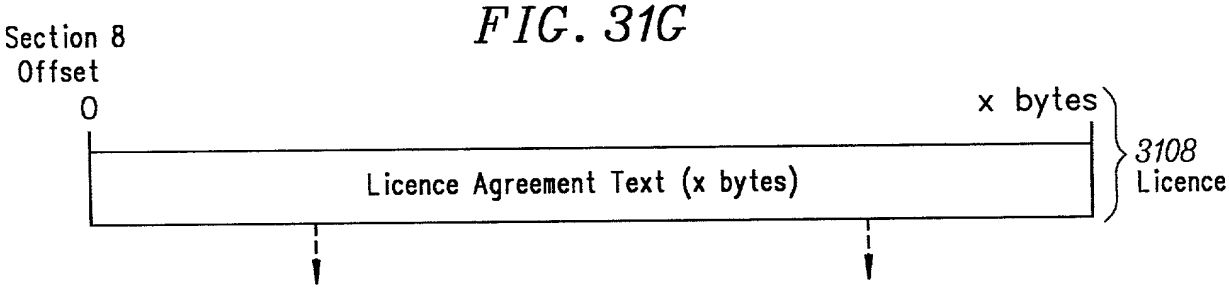


FIG. 31H

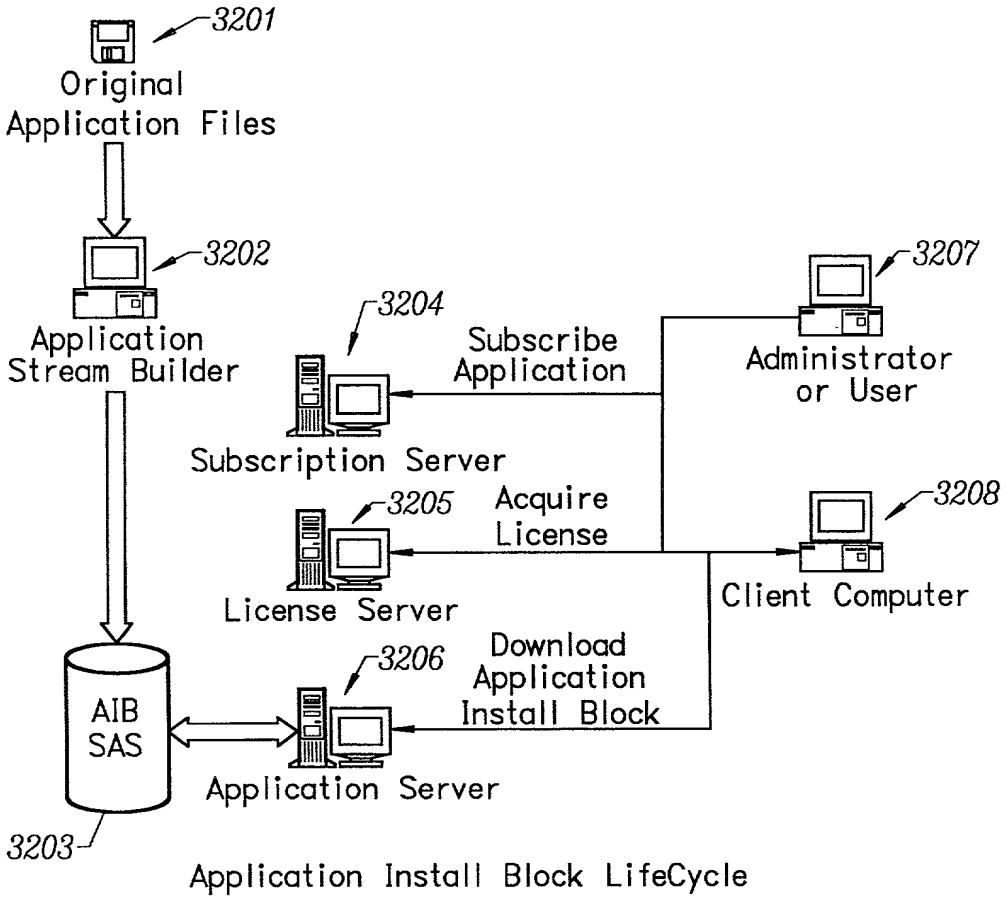
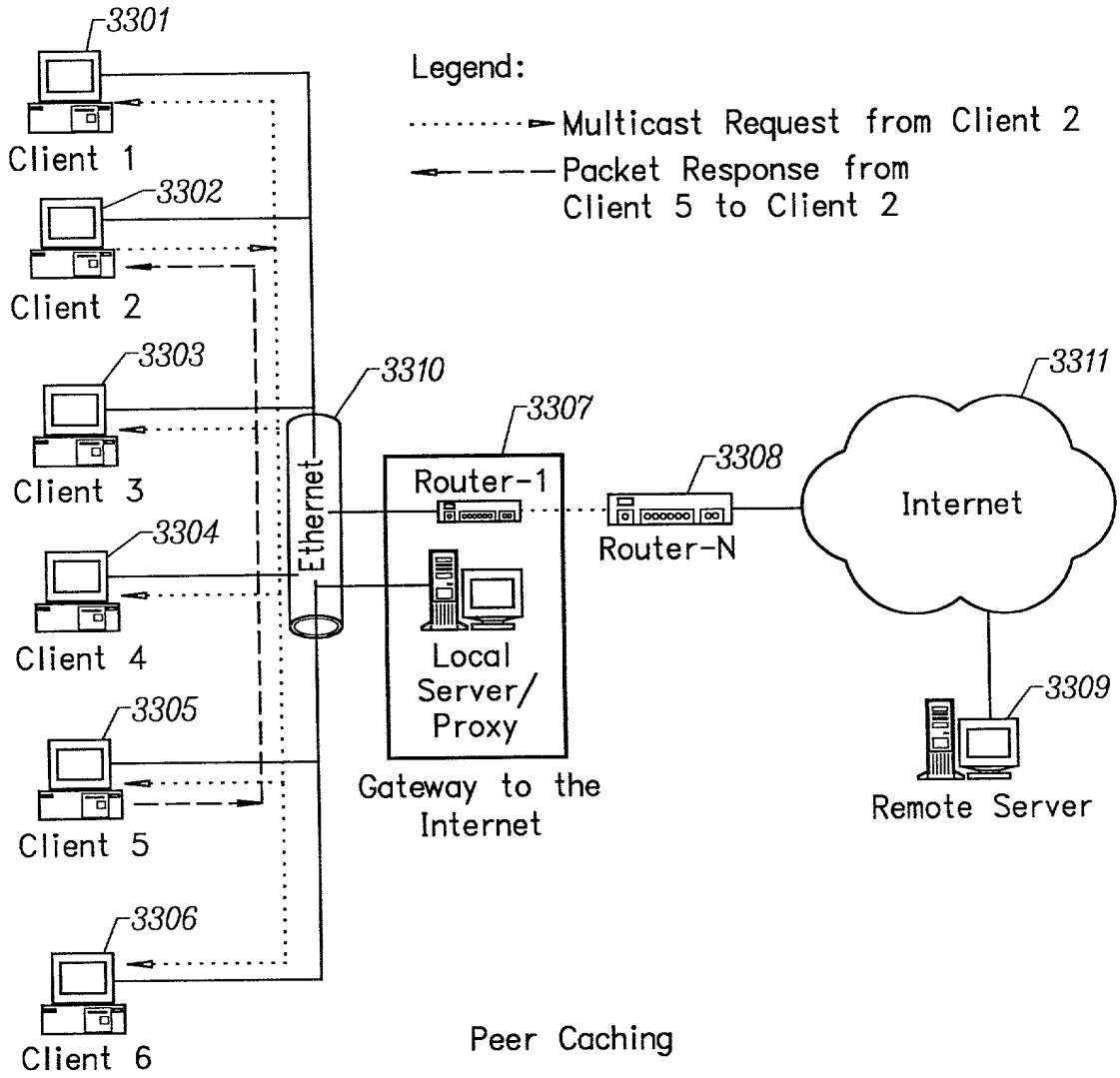
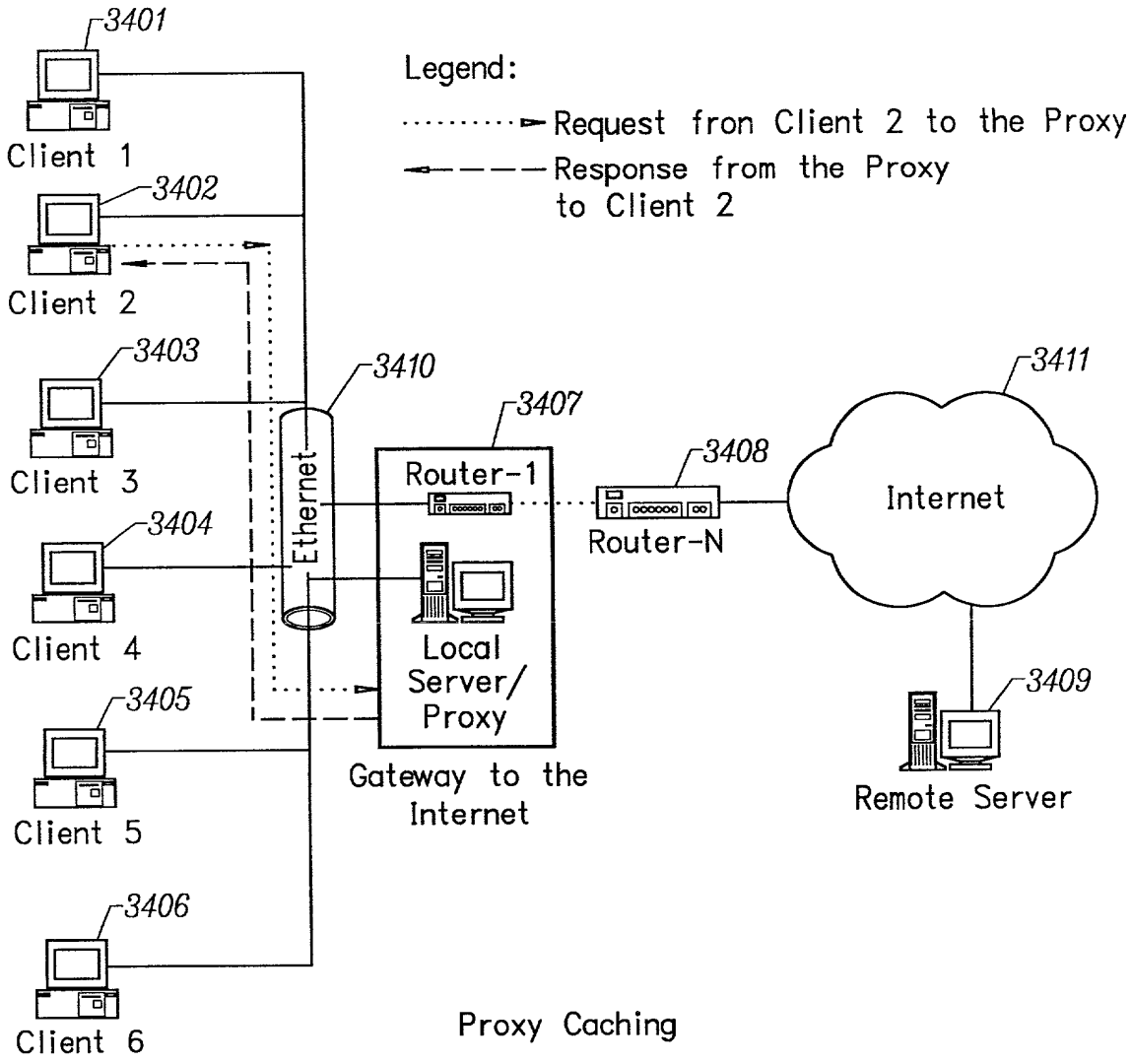

3201
Original
Application Files
3202
Application
Stream Builder
3204
Subscription Server
Subscribe
Application
3207
Administrator
or User
3205
License Server
Acquire
License
3208
Client Computer
AIB
SAS
3206
Application Server
Download
Application
Install Block
3203
Application Install Block LifeCycle

FIG. 32



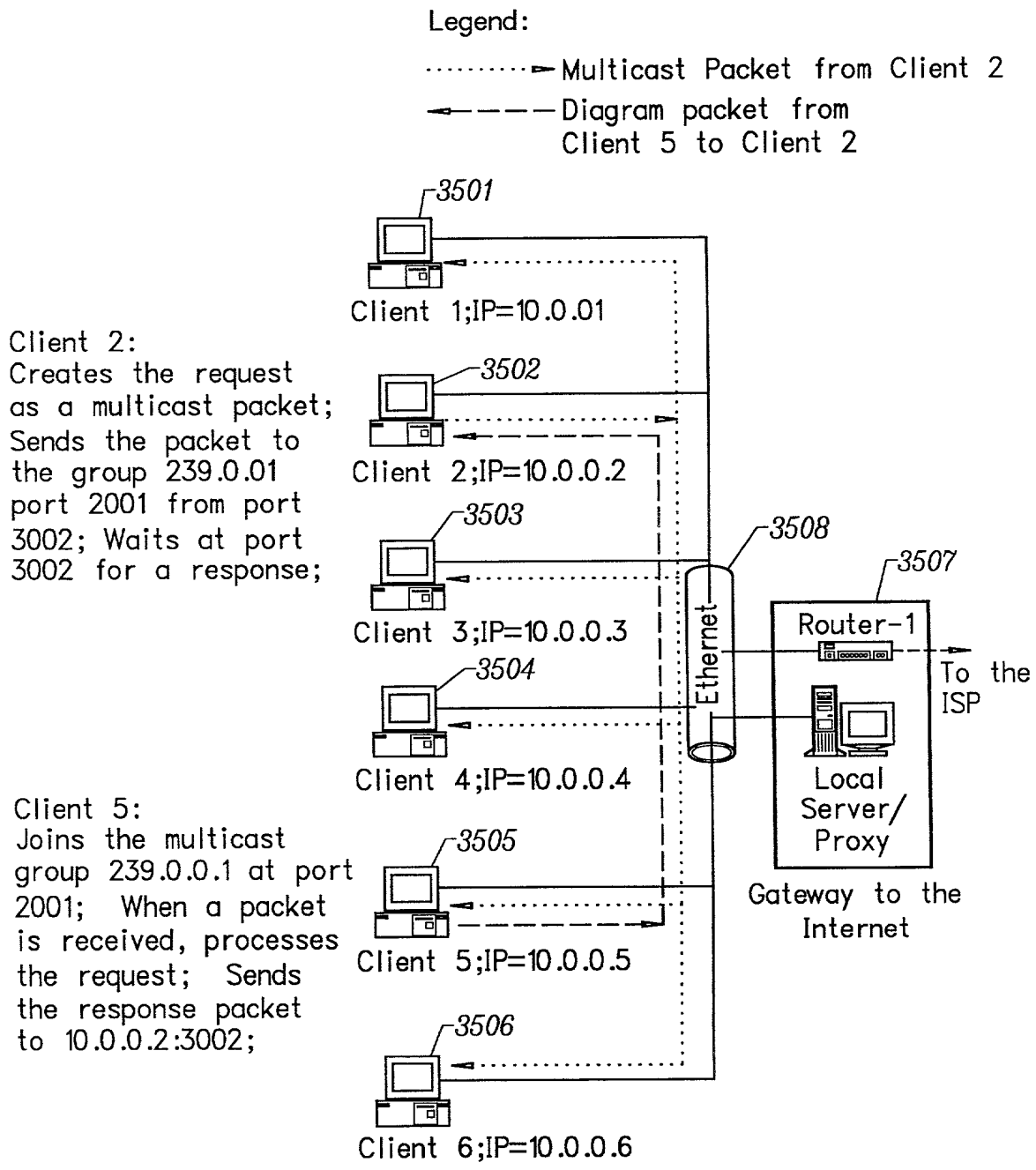
Peer Caching

FIG. 33



Proxy Caching

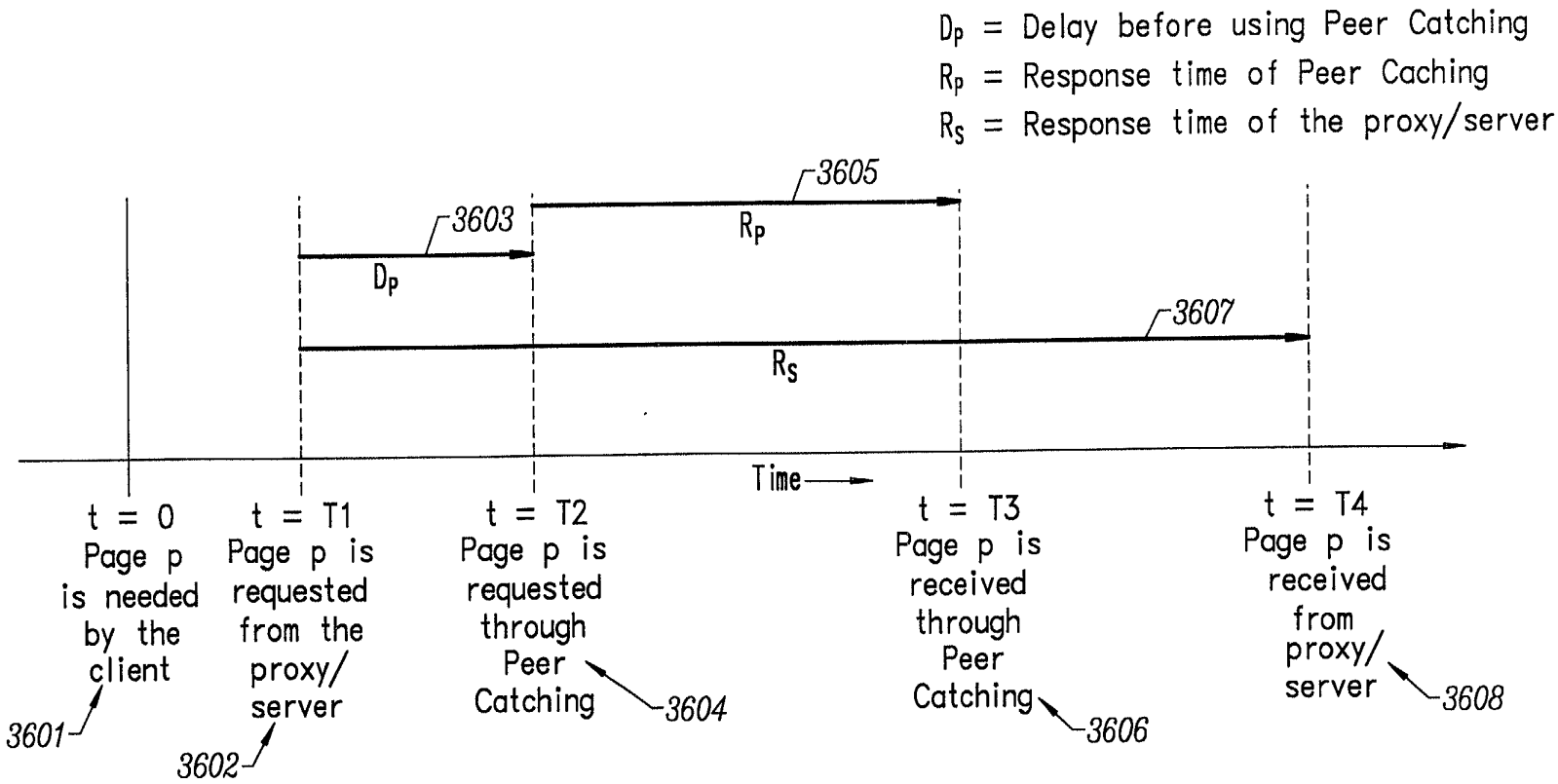
FIG. 34



Milticast within a LAN and Packet Protocol

FIG. 35

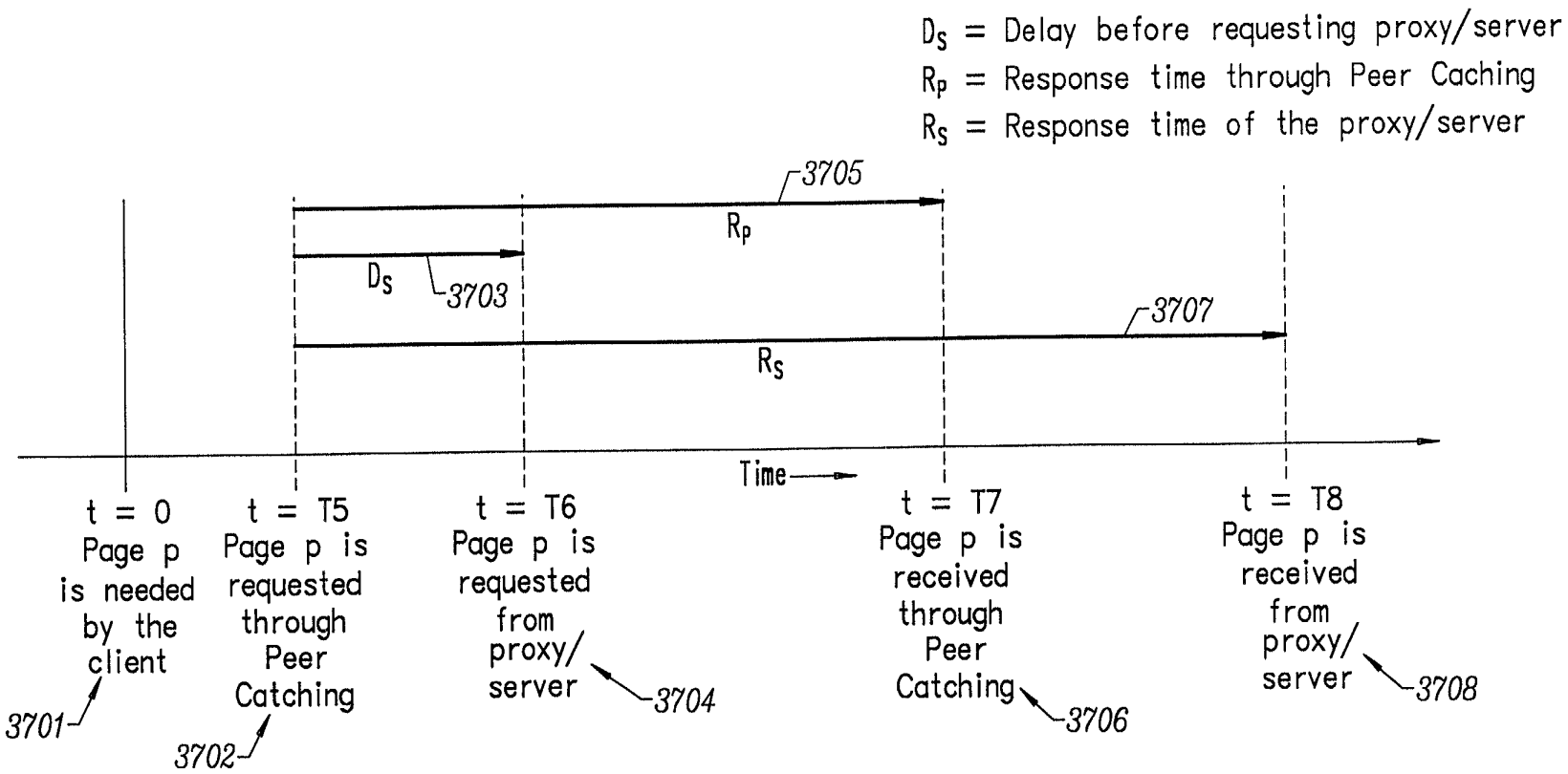
PCT/US03/00000



Concurrent Requesting - Proxy First

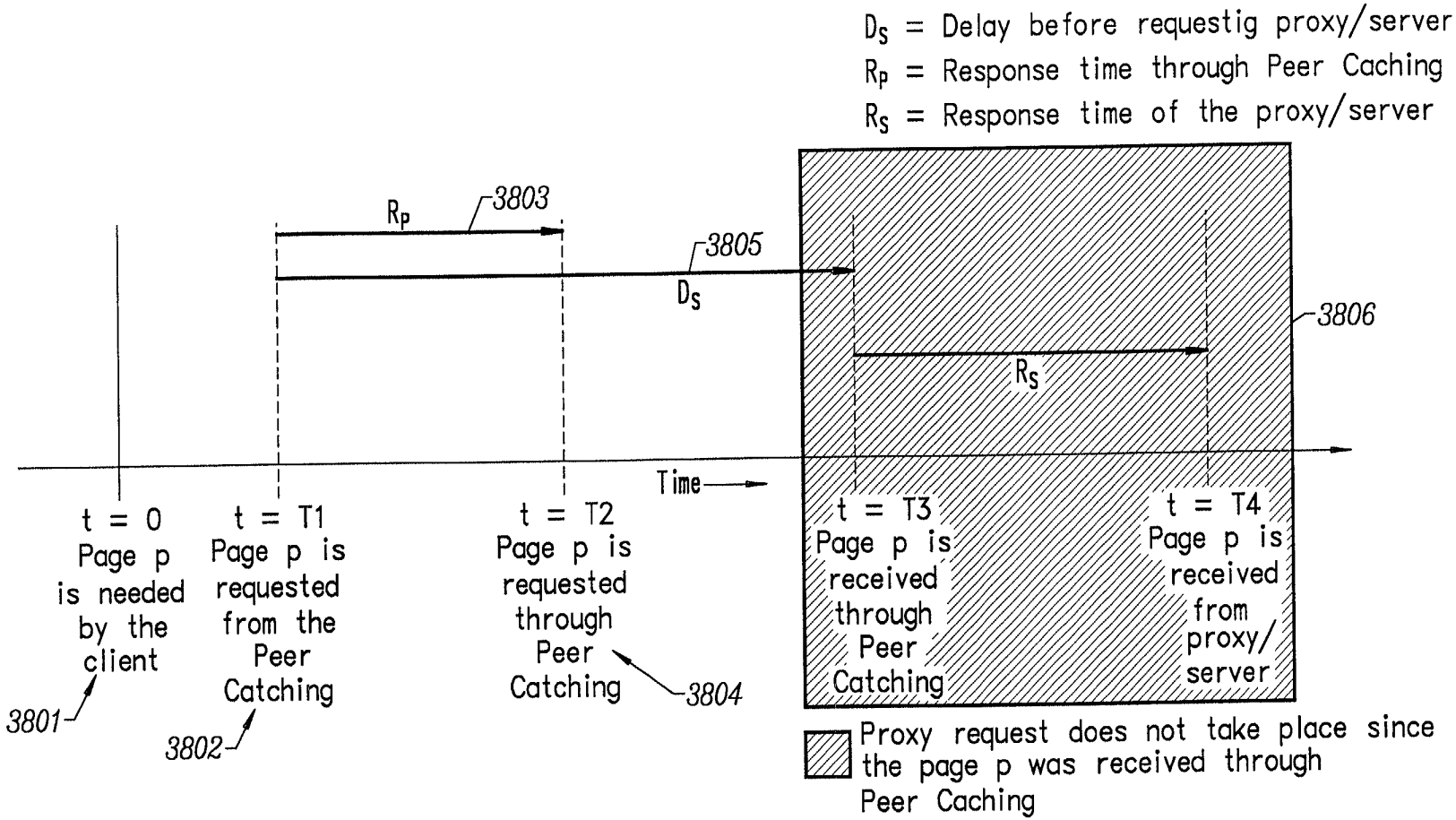
FIG. 36

FOUND DEDICATED



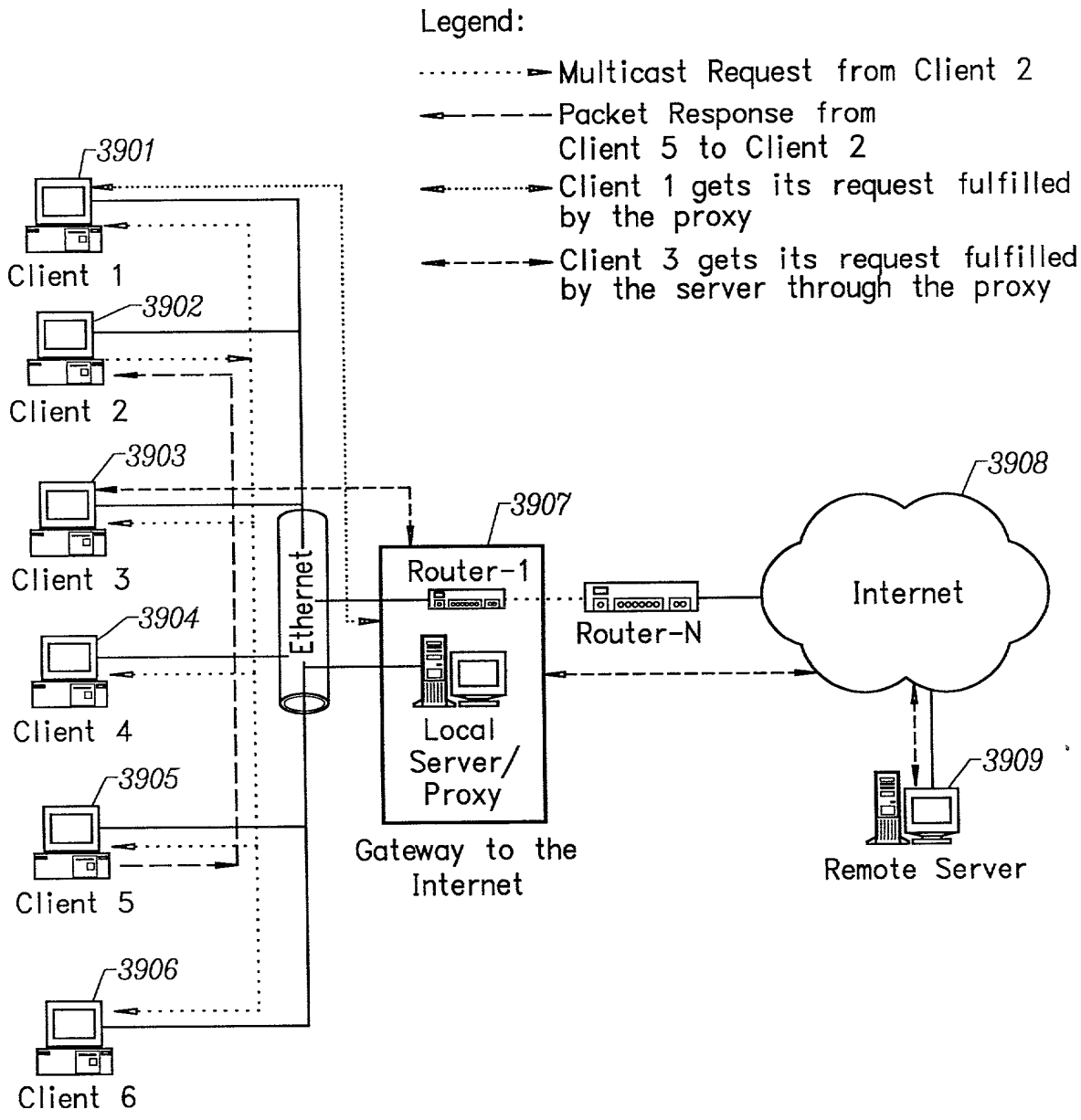
Concurrent Requesting - Peer Catching First

FIG. 37



Concurrent Requesting - Peer Caching Only

FIG. 38



Client-Server System with Peer and Proxy Caching

FIG. 39

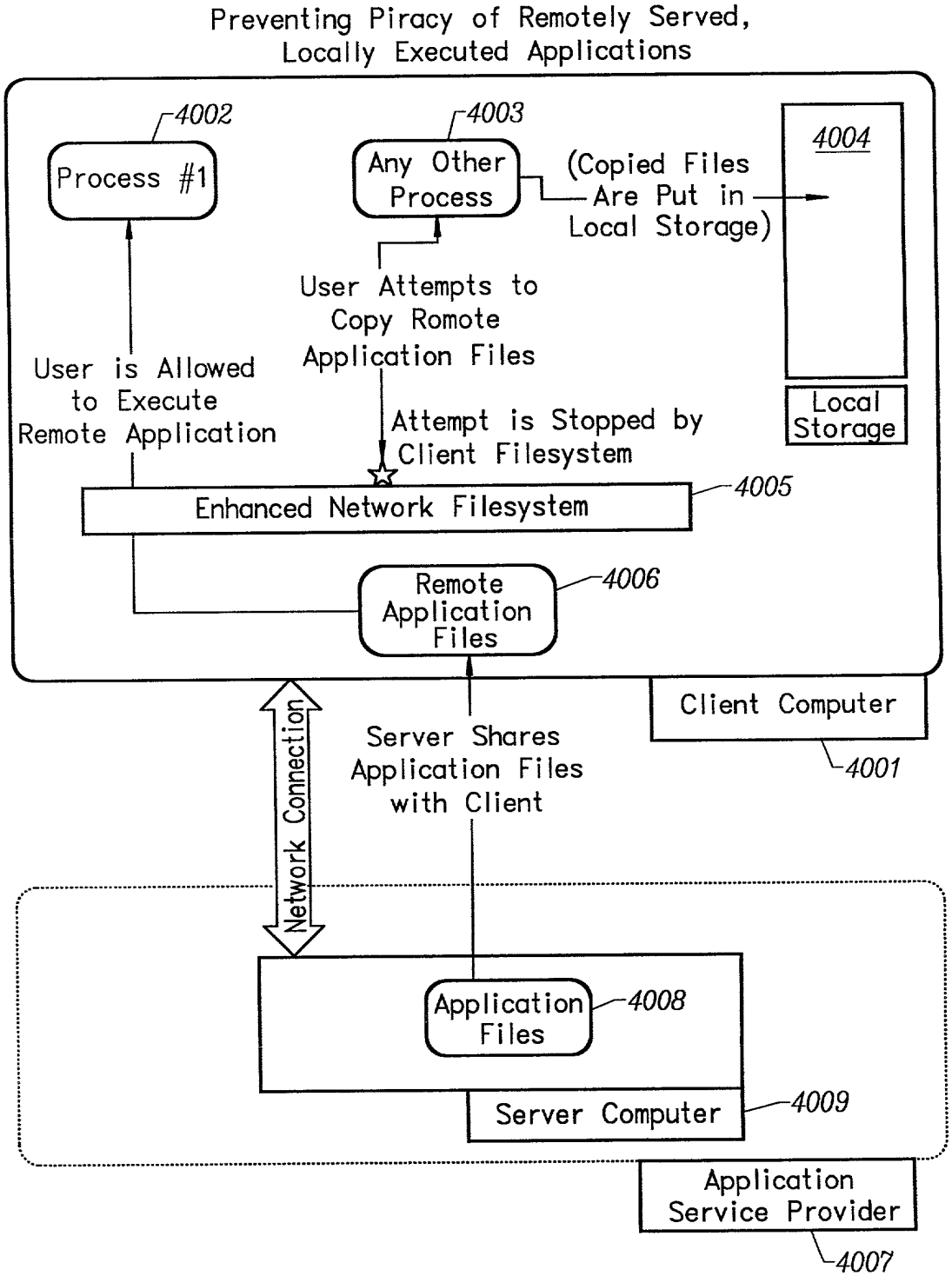

Preventing Piracy of Remotely Served, Locally Executed Applications
4002
Process #1
4003
Any Other Process
(Copied Files Are Put in Local Storage)
4004
Local Storage
User Attempts to Copy Romote Application Files
User is Allowed to Execute Remote Application
Attempt is Stopped by Client Filesystem
Enhanced Network Filesystem
4005
Remote Application Files
4006
Client Computer
4001
Network Connection
Server Shares Application Files with Client
Application Files
4008
Server Computer
4009
Application Service Provider
4007

FIG. 40

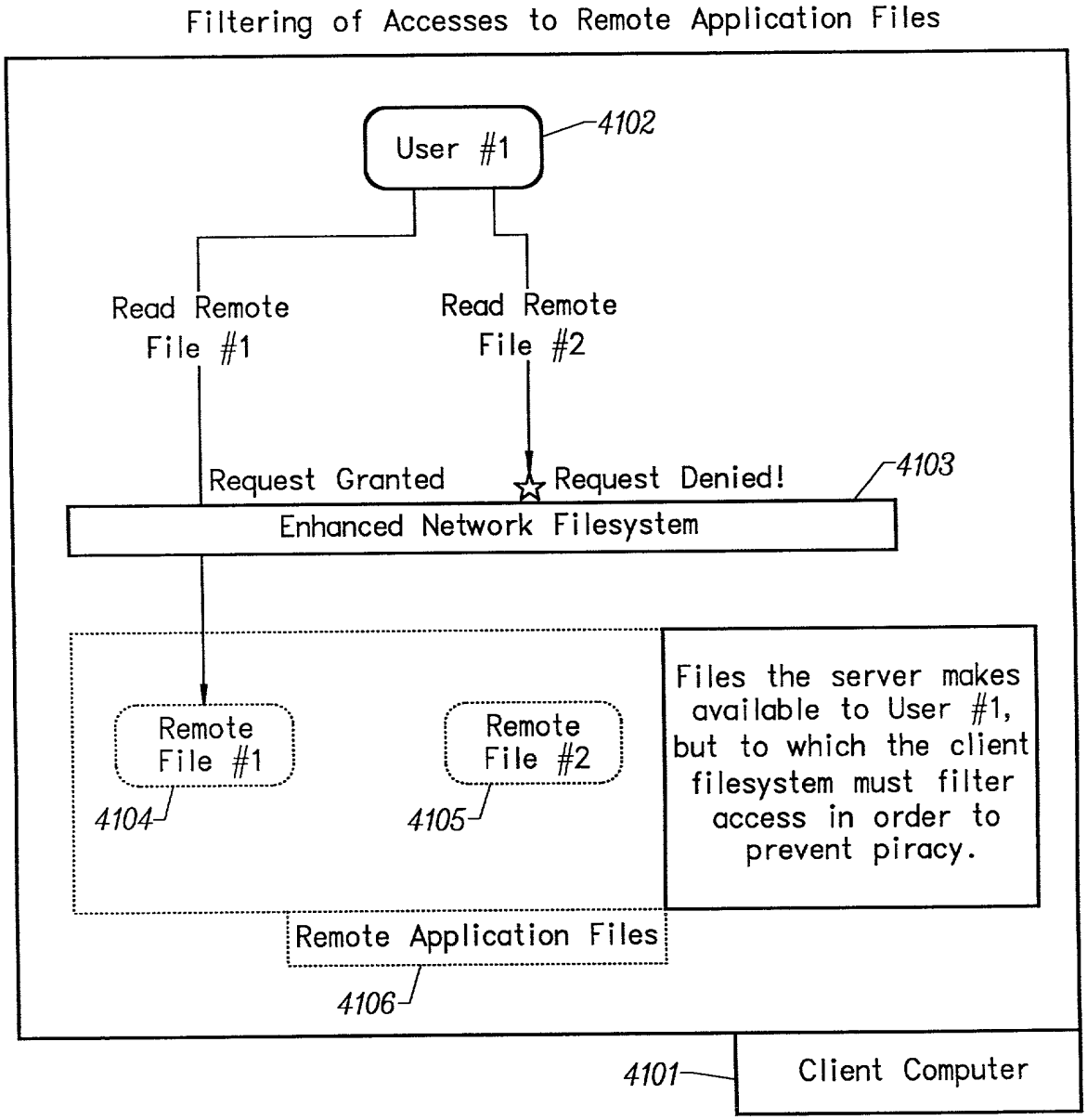

Filtering of Accesses to Remote Application Files
User #1
4102
Read Remote File #1
Read Remote File #2
Request Granted
Request Denied!
4103
Enhanced Network Filesystem
Remote File #1
4104
Remote File #2
4105
Files the server makes available to User #1, but to which the client filesystem must filter access in order to prevent piracy.
Remote Application Files
4106
4101
Client Computer

FIG. 41

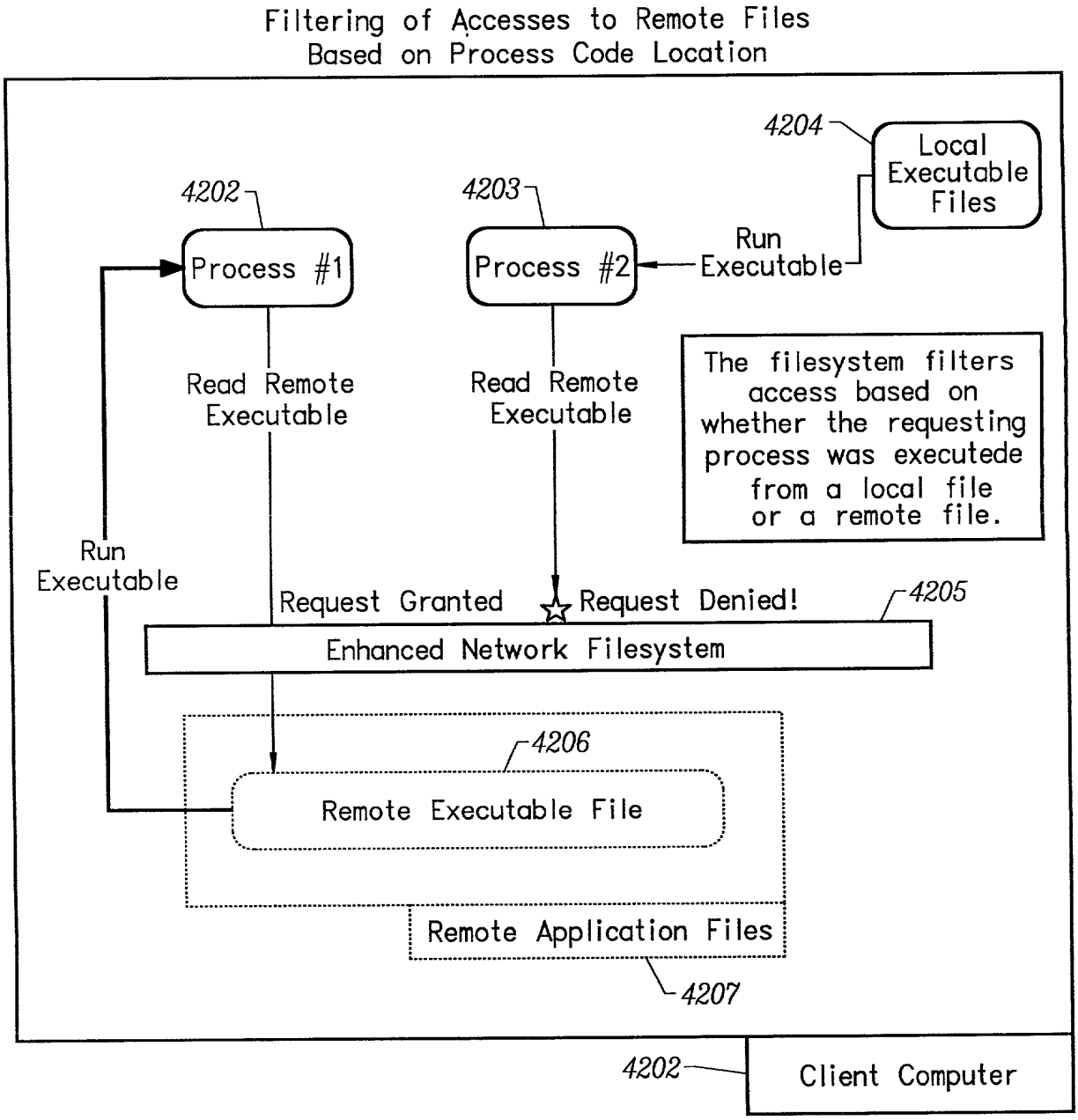

Filtering of Accesses to Remote Files
Based on Process Code Location
4204
Local Executable Files
4202
4203
Process #1
Process #2
Run Executable
Read Remote Executable
Read Remote Executable
The filesystem filters access based on whether the requesting process was executede from a local file or a remote file.
Run Executable
Request Granted
Request Denied!
4205
Enhanced Network Filesystem
4206
Remote Executable File
Remote Application Files
4207
4202
Client Computer

FIG. 42

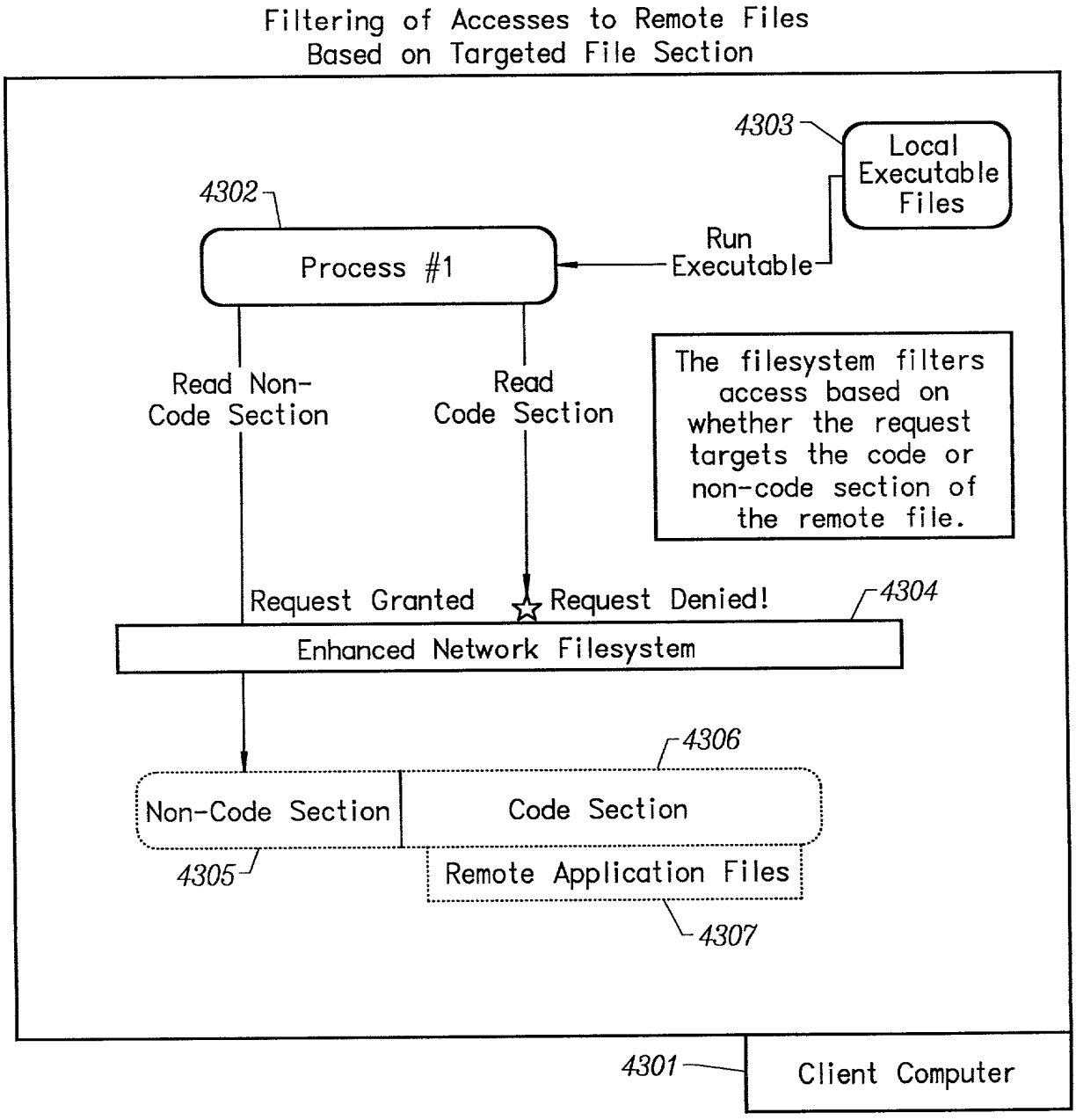

Filtering of Accesses to Remote Files
Based on Targeted File Section
4303
Local Executable Files
4302
Process #1
Run Executable
Read Non-Code Section
Read Code Section
The filesystem filters access based on whether the request targets the code or non-code section of the remote file.
Request Granted
Request Denied!
4304
Enhanced Network Filesystem
4306
Non-Code Section
Code Section
4305
Remote Application Files
4307
4301
Client Computer

FIG. 43

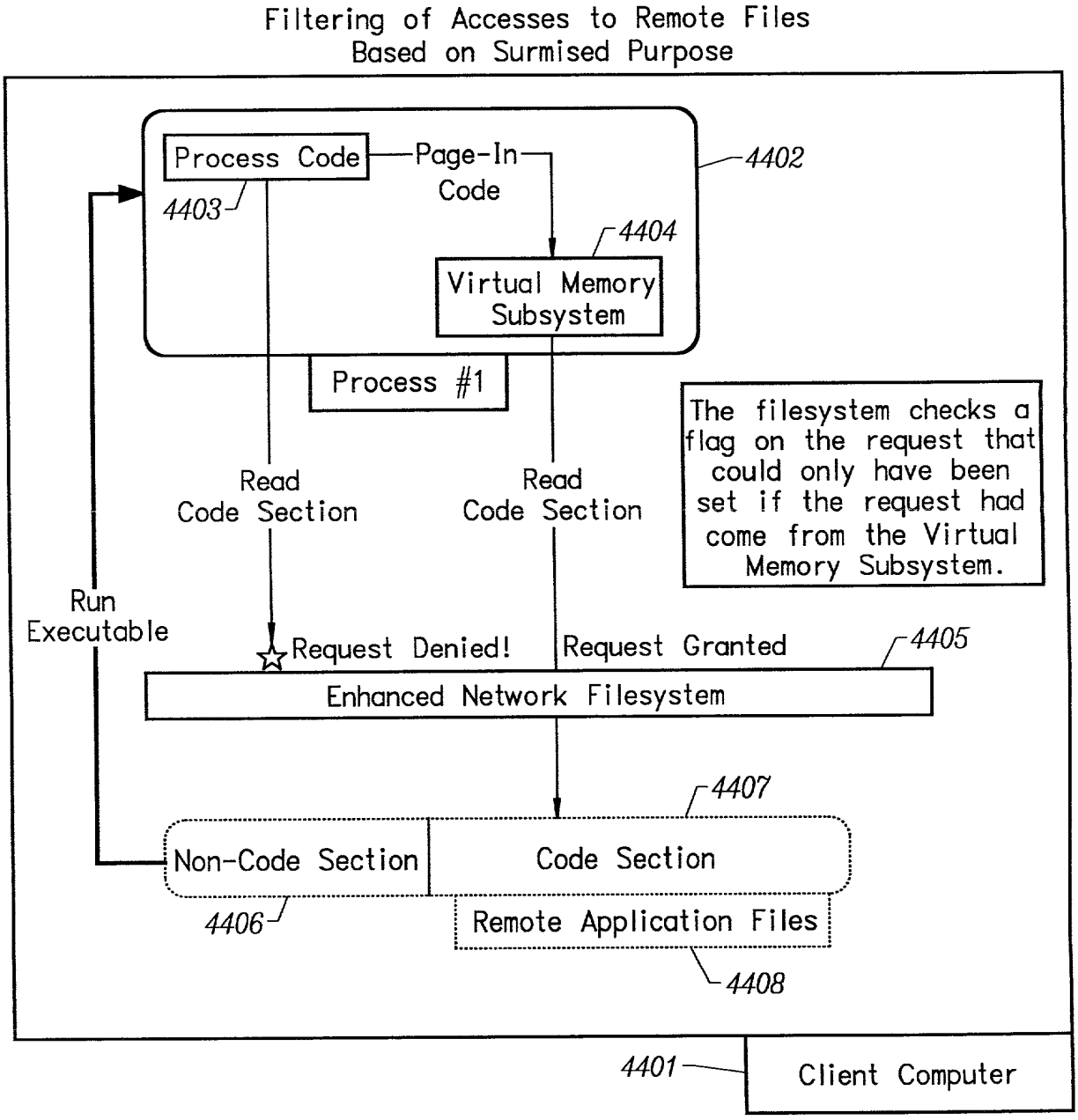

Filtering of Accesses to Remote Files
Based on Surmised Purpose
Process Code
Page-In
Code
4402
4403
4404
Virtual Memory
Subsystem
Process #1
The filesystem checks a flag on the request that could only have been set if the request had come from the Virtual Memory Subsystem.
Read
Code Section
Read
Code Section
Run
Executable
Request Denied!
Request Granted
4405
Enhanced Network Filesystem
4407
Non-Code Section
Code Section
4406
Remote Application Files
4408
4401
Client Computer

FIG. 44

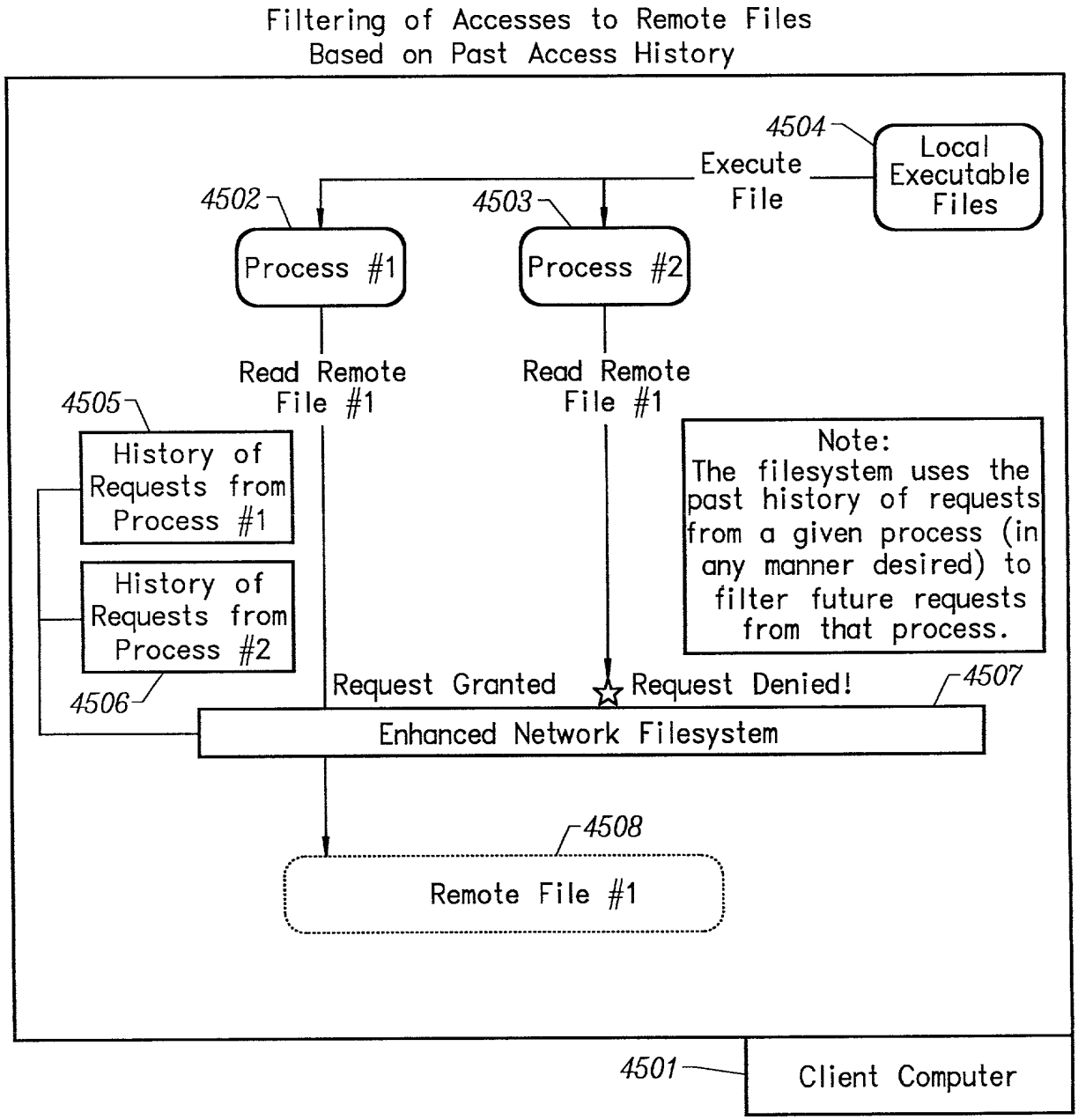

Filtering of Accesses to Remote Files
Based on Past Access History
4504
Local Executable Files
Execute File
4502
4503
Process #1
Process #2
Read Remote File #1
Read Remote File #1
4505
History of Requests from Process #1
History of Requests from Process #2
4506
Note:
The filesystem uses the past history of requests from a given process (in any manner desired) to filter future requests from that process.
Request Granted
Request Denied!
4507
Enhanced Network Filesystem
4508
Remote File #1
4501
Client Computer

FIG. 45